

ISI PAPER SERIES — PAPER 1

International Sovereignty Index (ISI)
Technical Specification
Version 1.0

International Sovereignty Institute — Research Unit

February 2026

Abstract

This document is the technical specification of the International Sovereignty Index (ISI), a composite indicator that measures the degree to which a sovereign state concentrates its critical external dependencies on a narrow set of counterparties. The ISI applies a Herfindahl–Hirschman framework across six thematic axes—financial sovereignty, energy dependency, technology and semiconductor dependency, defence industrial dependency, critical inputs and raw materials dependency, and logistics and freight dependency—each decomposed into one or two measurement channels. Country-level concentration scores are computed on the unit interval, arithmetically averaged into a single composite, and mapped to a four-tier classification (*highly concentrated*, *moderately concentrated*, *mildly concentrated*, *unconcentrated*). A deterministic scenario engine permits first-order sensitivity analysis by applying bounded multiplicative adjustments at the axis-score level. All computational

outputs are bound to a cryptographic integrity chain (SHA-256 hashing, Ed25519 signing) that enables third-party audit without access to the computation environment.

Beyond the core methodology, this specification formalises the determinism guarantees of the computation pipeline (ten invariants, floating-point assumptions, canonical serialisation rules, and prohibited operations), the snapshot materialisation architecture (directory structure, output schemas, methodology registry, and immutability contract), the read-only API contract (endpoint taxonomy, caching semantics, error codes, and backward-compatibility rules), the verification and test-coverage requirements (unit tests, property-based tests, golden-file tests, integration tests, and integrity-chain tests), a threat model with six adversary classes and four security goals, and a formal computational state model defining the ten-state pipeline with explicit irreversibility gates and mutation prohibitions.

ISI v 1.0 covers the EU-27. Extension to additional country sets requires source-level availability, harmonisation, and integrity validation. All scores are rounded to eight decimal places (round-half-to-even) prior to classification and hashing.

Keywords: sovereignty index, concentration measurement, composite indicator, Herfindahl–Hirschman, determinism, integrity chain

Contents

- 1 Introduction** **1**
 - 1.1 Scope 1
 - 1.2 Context 1
 - 1.3 Design Principles 1
 - 1.4 Document Map 2

- 2 Data Architecture** **2**
 - 2.1 Source Registry 2
 - 2.2 Coverage 2
 - 2.3 Data Availability Constraints (v 1.0) 3
 - 2.4 Ingestion Protocol 3
 - 2.5 Quality Controls 3

- 3 Mathematical Framework** **4**
 - 3.1 Notation 4
 - 3.2 Concentration Formula 4
 - 3.3 Cross-Channel Aggregation 5
 - 3.4 Category-Weighted Channel B 5
 - 3.5 Single-Channel Fuel Averaging 5
 - 3.6 Composite Score 5
 - 3.7 Rounding 6
 - 3.8 Classification Thresholds 6

- 4 Axis Specifications** **6**
 - 4.1 Axis 1: Financial Sovereignty 6
 - 4.2 Axis 2: Energy Dependency 7
 - 4.3 Axis 3: Technology and Semiconductor Dependency 8
 - 4.4 Axis 4: Defence Industrial Dependency 8
 - 4.5 Axis 5: Critical Inputs / Raw Materials 9
 - 4.6 Axis 6: Logistics and Freight Dependency 10
 - 4.7 Axis Summary 10

- 5 Aggregation and Classification** **10**
 - 5.1 Aggregation Pipeline 10
 - 5.2 Missing-Axis Protocol 11
 - 5.3 Ranking Methodology 11
 - 5.4 Vintage and Periodicity 11

- 6 Scenario Engine** **12**
 - 6.1 Purpose 12

6.2	Multiplicative Adjustment Model	12
6.3	Composite Recomputation	12
6.4	Rank Recomputation	12
6.5	Constraints	13
6.6	Use Cases	13
6.7	Limitations of the Scenario Engine	13
6.8	Isolation Guarantee	13
7	Integrity Chain	14
7.1	Design Rationale	14
7.2	Country-Level Hash	14
7.3	Snapshot Hash	15
7.4	Digital Signature	15
7.5	Published Artefacts	15
7.6	Verification Procedure	15
7.7	Key Management	16
8	Limitations	16
9	Governance and Versioning	17
9.1	Custodian	17
9.2	Versioning Scheme	17
9.3	Change Protocol	19
9.4	Data-Revision Policy	19
9.5	Publication Schedule	19
9.6	Archival	19
10	Determinism and Computational Guarantees	20
10.1	Scope of Determinism	20
10.2	Formal Invariants	20
10.3	Floating-Point Assumptions	21
10.4	Canonical Serialisation	22
10.5	Prohibited Operations	23
10.6	Hash Dependency Graph	24
10.7	Determinism Certification	24
11	Snapshot Materialisation and Registry Architecture	24
11.1	Design Principles	25
11.2	Directory Structure	25
11.3	Output Schema	25
11.3.1	Country Scores (country_scores.csv)	25
11.3.2	Axis Scores (axis_scores.csv)	26

11.3.3 Channel Scores (<code>channel_scores.csv</code>)	26
11.4 Methodology Registry	27
11.5 Snapshot Metadata	28
11.6 Immutability Contract	28
11.7 Manifest Construction	29
11.8 Pipeline Architecture	29
11.9 Hash Cascade Logic	30
11.10 Backward Compatibility of the Directory Schema	31
12 API Contract and Serving Guarantees	31
12.1 Transport and Protocol	31
12.2 Endpoint Taxonomy	32
12.2.1 GET <code>/v1/snapshots</code>	32
12.2.2 GET <code>/v1/snapshots/{vintage}/scores</code>	32
12.2.3 GET <code>/v1/snapshots/{vintage}/axes</code>	32
12.2.4 GET <code>/v1/snapshots/{vintage}/channels</code>	33
12.2.5 GET <code>/v1/snapshots/{vintage}/integrity</code>	33
12.2.6 GET <code>/v1/snapshots/{vintage}/registry</code>	33
12.2.7 GET <code>/v1/scenario</code>	33
12.3 ETag and Conditional Requests	33
12.4 Error Taxonomy	34
12.5 Rate Limiting	34
12.6 Versioned Endpoint Prefix	35
12.7 Backward-Compatibility Rules	35
12.8 Content Negotiation	35
12.9 Health and Readiness	36
12.10 Cross-Origin Resource Sharing	36
13 Verification, Test Coverage, and Formal Assurance	36
13.1 Test Infrastructure	36
13.2 Unit Tests	37
13.3 Property-Based Tests	38
13.4 Golden-File Tests	40
13.5 Integration Tests	41
13.6 Integrity-Chain Tests	43
13.7 Coverage Requirements	44
13.8 Regression Protocol	44
13.9 Continuous Integration	45
13.10 Formal Assurance Statement	45
14 Threat Model and Trust Assumptions	46

14.1 Adversary Classes	46
14.2 Security Goals	46
14.3 Out-of-Scope Threats	47
14.4 Trust Anchors	47
14.5 Mitigation Matrix	48
14.6 Key Rotation and Revocation Policy	48
14.7 Security Guarantees and Cryptographic Assumptions	49
15 Formal Computational State Model	51
15.1 State Definitions	51
15.2 Transition Functions	52
15.3 Irreversibility Points	53
15.4 State Mutation Prohibition	53
15.5 Scenario Engine as Non-State-Altering Branch	54
15.6 Deterministic Transition Constraints	54
15.7 Axis-Completeness Invariant	55
15.8 State Diagram Summary	55
16 System Architecture Overview	56
A Axis Registry	59
B Classification Thresholds	59
C Warning Code Registry	59

1 Introduction

1.1 Scope

The International Sovereignty Index (ISI) quantifies the structural concentration of a country’s critical external dependencies. It does not measure sovereignty in the normative or legal sense. It measures the extent to which a state relies on a small number of foreign partners for essential goods, services, and capabilities across six domains: financial market access, energy supply, technology and semiconductor inputs, defence-industrial procurement, critical raw materials, and logistics and freight infrastructure.

The index operates at the country–year level. Each observation produces a composite score on the unit interval $[0, 1]$ and a categorical classification. Higher values denote greater concentration and, by implication, greater structural exposure to partner-specific disruption.

ISI v 1.0 covers the EU-27. The computation pipeline, integrity chain, and classification logic are country-set agnostic. Extension to non-EU countries is conditional on source-level data availability, cross-source harmonisation, and integrity validation for each additional jurisdiction.

1.2 Context

Existing composite indicators in the sovereignty-adjacent space—the Gygli et al. [1] Globalisation Index, the UNCTAD [2] Productive Capacities Index, and the EU Global Gateway scorecards—measure either *openness* or *capacity*. None directly measures *concentration* of external dependence. The ISI occupies a distinct methodological position: it quantifies the structural geometry of dependence, complementing—not substituting—openness and capacity measures.

1.3 Design Principles

Five principles constrain the index architecture:

- P1. Transparency.** Every formula, data source, and threshold is published. No proprietary model or hidden weighting layer exists.
- P2. Reproducibility.** Given identical input data, any competent third party must be able to reproduce the published scores to full numerical precision.
- P3. Auditability.** Cryptographic integrity records (hash chains and digital signatures) accompany every published snapshot, enabling verification without access to the original computation environment.
- P4. Parsimony.** The index uses exactly one concentration measure, the Herfindahl–Hirschman Index, applied uniformly across all axes. No axis receives preferential weighting in the composite.
- P5. Conservatism.** Where methodological choices are ambiguous, the index defaults to the option that is simpler, more established, and easier to audit.

1.4 Document Map

The remainder of this document is organised as follows. Section 2 describes the data sources, ingestion protocol, and quality controls. Section 3 presents the mathematical framework—the concentration formula, rounding rules, and classification thresholds. Section 4 specifies each of the six thematic axes. Section 5 details the composite aggregation and ranking procedure. Section 6 documents the scenario simulation model. Section 7 describes the cryptographic integrity mechanism. Section 8 enumerates the structural limitations of the index. Section 9 sets out the governance and versioning protocol. Section 10 formalises the determinism guarantees and computational invariants of the pipeline. Section 11 specifies the snapshot materialisation architecture, output schemas, and immutability contract. Section 12 defines the read-only API contract, endpoint taxonomy, and backward-compatibility rules. Section 13 documents the verification test suite and coverage requirements. Section 14 defines the threat model, adversary classes, security goals, trust anchors, and key-rotation policy. Section 15 presents a formal computational state model that specifies the ten pipeline states, nine transition functions, two irreversibility gates, and the scenario engine’s isolation as a non-state-altering branch. Section 16 provides a concise four-layer system abstraction summarising the mathematical, computational, integrity, and governance layers. Three reference appendices follow.

2 Data Architecture

2.1 Source Registry

The ISI draws on five primary statistical sources, each mapped to one or more thematic axes. Table 1 summarises the registry.

Table 1. Primary data sources (v 1.0).

Axis	Source	Granularity	Window
1—Financial	BIS LBS / IMF CPIS	Counterparty, quarterly	2024-Q4
2—Energy	Eurostat <code>nrg_ti_*</code>	Fuel, partner, annual	2024
3—Tech./Semi.	Eurostat Comext ds-045409	CN8, bilateral, annual	2022–2024
4—Defence	SIPRI ATD (TIV)	System, bilateral, annual	2019–2024
5—Crit. Inputs	Eurostat Comext CN8	CN8, bilateral, annual	2022–2024
6—Logistics	Eurostat transport stats	Mode, partner, annual	2022–2024

2.2 Coverage

The unit of observation is the country–year pair (i, t) . The reference population for ISI v 1.0 is the EU-27. The computation pipeline, integrity chain, and classification logic are country-set agnostic. Extension to non-EU countries is conditional on source-level data availability, cross-source harmonisation, and integrity validation for each additional jurisdiction.

For a given vintage year t , a country i is included in the published snapshot if and only if at least four of the six axes yield a computable concentration score. Where fewer than four axes are computable, the country is flagged with warning code $W\text{-COV}$ and excluded from the ranked output (see Section C).

2.3 Data Availability Constraints (v 1.0)

Only ISI 2024 is materialisable under v 1.0. The binding constraints are:

- **BIS LBS.** The Locational Banking Statistics are published with a two-quarter lag. Only the 2024-Q4 release provides sufficient counterparty coverage for the EU-27. Earlier quarters lack the partner disaggregation required for Channel A of Axis 1.
- **Eurostat energy.** The `nrg_ti_*` tables report annual fuel-level import data. The 2024 vintage is the most recent complete release.
- **SIPRI.** The Arms Transfers Database reports calendar-year deliveries. The six-year window (2019–2024) is the minimum required to smooth lumpy transfer patterns for EU member states.
- **Eurostat transport.** Modal freight statistics lag by one to two years. The 2022–2024 window represents the best available coverage at time of computation.

An ISI 2023 vintage is infeasible: BIS counterparty data for 2023-Q4 lack sufficient EU-27 partner coverage, and the SIPRI window would contract below the minimum smoothing threshold for several member states. Time-series extension requires additional source vintages and is not supported in v 1.0.

2.4 Ingestion Protocol

Data ingestion follows a four-stage pipeline:

1. **Acquisition.** Raw files are retrieved from the source API or bulk-download portal in the format published by the provider (CSV, SDMX, JSON).
2. **Parsing.** Source-specific parsers normalise records into a common schema per axis. Each parser is a dedicated module in the backend pipeline.
3. **Validation.** Row-level checks enforce non-negative values, valid ISO 3166-1 alpha-2 codes (EU member states), and year-range bounds. Records that fail validation are quarantined and logged.
4. **Loading.** Validated records are loaded into the computation layer. No post-load transformation occurs outside the formula definitions in Section 3.

2.5 Quality Controls

Two quality controls apply at the axis level:

Temporal gap detection.

If the most recent available data point for axis a and country i predates the snapshot vintage year t by more than two years, the axis score is computed from the latest available year and warning code `W-LAG` is attached.

Channel fallback basis.

Where an axis defines two channels but only one is computable for a given country, the axis score reverts to the surviving channel. The fallback basis (`BOTH`, `A_ONLY`, `B_ONLY`, `NONE`) is recorded in the output metadata per country–axis pair and is included in the country-level hash of the integrity chain (Section 7.2).

Warning codes are propagated to the published output but do not alter the numerical score. They serve as metadata for downstream consumers.

3 Mathematical Framework

This section defines the core formulae that transform raw bilateral flow data into concentration scores on the unit interval. All notation introduced here applies throughout the remainder of the document.

3.1 Notation

Let i index countries ($i = 1, \dots, N$; $N = 27$ for v 1.0). Let a index thematic axes ($a = 1, \dots, 6$). Where an axis defines two channels, they are denoted $\text{ch} \in \{A, B\}$. Not all axes use a two-channel decomposition; axis-specific structures are detailed in Section 4.

For a given country i , axis a , and channel ch , the partner-share vector is $\mathbf{s}_i^{(\text{ch})} = (s_{i,1}^{(\text{ch})}, \dots, s_{i,J}^{(\text{ch})})$, where $s_{i,j}^{(\text{ch})}$ is the share of partner j in the total flow and $\sum_j s_{i,j}^{(\text{ch})} = 1$.

3.2 Concentration Formula

Concentration is measured by the Herfindahl–Hirschman Index applied to the partner-share vector. For country i on channel ch :

$$C_i^{(\text{ch})} = \sum_{j=1}^J (s_{i,j}^{(\text{ch})})^2 \quad (1)$$

The resulting value lies on the interval $(0, 1]$. A value of 1 indicates total dependence on a single partner; values approaching 0 indicate dispersion across many partners.

Relation to the classical HHI. The traditional HHI in industrial-organisation literature is computed on percentage shares and ranges from 0 to 10 000. The ISI operates on fractional shares, yielding a normalised HHI on $(0, 1]$. The two forms are related by a factor of 10 000.

3.3 Cross-Channel Aggregation

For axes that define two channels, the axis-level score is the arithmetic mean of the channel-level scores:

$$M_i^{(a)} = 0.5 C_i^{(A)} + 0.5 C_i^{(B)} \quad (2)$$

Equal weighting is a deliberate design choice (Principle **P4**). No empirical basis exists for assigning asymmetric importance to supply versus destination concentration in a general-purpose index.

Where only one channel is computable (fallback basis A_ONLY or B_ONLY), the axis score equals the surviving channel score.

3.4 Category-Weighted Channel B

For axes where Channel *B* encompasses multiple product sub-categories or capability blocks (see Section 4), a category-weighted variant is used. Let $k = 1, \dots, K$ index the sub-categories and let $V_i^{(k)}$ denote the total flow value of sub-category k for country i . Then:

$$C_i^{(B)} = \frac{\sum_{k=1}^K C_i^{(B,k)} V_i^{(k)}}{\sum_{k=1}^K V_i^{(k)}} \quad (3)$$

where $C_i^{(B,k)}$ is the HHI of sub-category k . This ensures that small-volume sub-categories do not dominate the axis score.

3.5 Single-Channel Fuel Averaging

Axis 2 (Energy) does not use a two-channel decomposition. Instead, a per-fuel HHI is computed for each fossil-fuel category and the axis score is the arithmetic average across fuels:

$$M_i^{(2)} = \frac{1}{|\mathcal{F}_i|} \sum_{f \in \mathcal{F}_i} C_i^{(f)} \quad (4)$$

where \mathcal{F}_i is the set of fuel categories for which country i has non-zero imports (see Section 4.2).

3.6 Composite Score

The composite ISI score for country i is the unweighted arithmetic mean of the six axis-level scores:

$$\text{ISI}_i = \frac{A_1 + A_2 + A_3 + A_4 + A_5 + A_6}{6} \quad (5)$$

where $A_a \equiv M_i^{(a)}$ for notational compactness. The composite inherits the unit-interval range of the axis scores.

Equal weighting. No differential axis weights are applied. Differential weighting would require an empirical or normative justification for asserting that one domain matters more than another for *all* countries at *all* times. No credible basis for such an assertion exists. The equal-weight rule is transparent, reproducible, and neutral across policy domains.

3.7 Rounding

All published scores—axis-level and composite—are rounded to eight decimal places using the *round-half-to-even* convention (IEEE 754 banker’s rounding). The rounding precision is set by the backend constant `ROUND_PRECISION = 8`. Rounding is applied *before* classification and *before* hashing. Intermediate calculations retain full floating-point precision.

3.8 Classification Thresholds

The rounded composite score is mapped to a four-tier ordinal classification:

Table 2. Classification thresholds.

Composite range	Classification
$ISI_i \geq 0.50$	highly_concentrated
$0.25 \leq ISI_i < 0.50$	moderately_concentrated
$0.15 \leq ISI_i < 0.25$	mildly_concentrated
$ISI_i < 0.15$	unconcentrated

The threshold values represent a proportional transposition of the US Department of Justice / FTC Horizontal Merger Guidelines HHI bands [3] to the fractional scale, with an additional tier boundary at 0.15 introduced to separate mildly concentrated from unconcentrated states. They are defined in the version-controlled methodology registry and enforced by a single classification function in the classification module. The thresholds are fixed parameters of `ISI v 1.0`. Any modification to the threshold values constitutes a major-version increment under the governance protocol (Section 9).

4 Axis Specifications

This section specifies each of the six thematic axes as implemented in the `ISI v 1.0` backend. For every axis, the specification covers: (i) the economic concept measured, (ii) the data source and product/flow scope, (iii) the channel decomposition and aggregation variant, (iv) the fallback-basis logic, and (v) boundary conditions.

4.1 Axis 1: Financial Sovereignty

Concept. Concentration of a country’s cross-border financial exposure—banking claims and portfolio debt—across counterparty countries.

Sources.

- **Channel A:** BIS Locational Banking Statistics (Table A6.2, by counterparty country). Quarterly frequency; the most recent quarter available for the vintage year is used (typically Q4).
- **Channel B:** IMF Coordinated Portfolio Investment Survey (CPIS), debt securities. Annual frequency.

Channel decomposition.

- **Channel A (banking claims):** Shares of inward banking claims on country i by creditor country. $s_{i,j}^{(A)} = \text{claims}_{j \rightarrow i} / \text{claims}_{\cdot \rightarrow i}$.
- **Channel B (portfolio debt):** Shares of country i 's outward portfolio-debt holdings by debtor country. $s_{i,j}^{(B)} = \text{debt}_{i \rightarrow j} / \text{debt}_{i \rightarrow \cdot}$.

Cross-channel aggregation. Arithmetic mean (Equation (2)).

Fallback basis. When both channels are available: BOTH. When only BIS data exist: A_ONLY. When only CPIS data exist: B_ONLY. When neither source reports data for a country: NONE; axis score is null.

Boundary conditions. Countries absent from the BIS reporting population receive $C_i^{(A)} = \text{null}$. Countries absent from the CPIS receive $C_i^{(B)} = \text{null}$. If both channels are null the axis score is null and flagged W-ZER.

4.2 Axis 2: Energy Dependency

Concept. Concentration of a country's fossil-fuel energy imports across partner countries.

Source. Eurostat energy tables (nrg_ti_*). Annual frequency. Scope: natural gas, crude oil, and solid fossil fuels.

Aggregation variant. Axis 2 does *not* use a two-channel decomposition. It produces a single score via fuel averaging. A per-fuel HHI is computed for each fossil-fuel category with non-zero imports, and the axis score is the arithmetic average across those fuels (Equation (4)):

$$M_i^{(2)} = \frac{1}{|\mathcal{F}_i|} \sum_{f \in \mathcal{F}_i} C_i^{(f)}$$

where $\mathcal{F}_i \subseteq \{\text{gas, oil, solid fossil}\}$ is the set of fuel types for which country i has non-zero imports.

Boundary conditions. If a country records zero fossil-fuel imports across all three categories, the axis score is null and flagged `W-ZER`; the country is then subject to the missing-axis coverage rule (Section 5.2). If imports exist for only one or two fuel categories, the denominator $|\mathcal{F}_i|$ adjusts to the number of non-zero categories.

4.3 Axis 3: Technology and Semiconductor Dependency

Concept. Concentration of a country’s imports of semiconductor devices and electronic components across partner countries.

Source. Eurostat Comext (dataset `ds-045409`). Product scope: HS 8541 (semiconductor devices) and HS 8542 (electronic integrated circuits).

Channel decomposition.

- **Channel A (aggregate):** HHI of partner-country shares across combined HS 8541 + 8542 imports.
- **Channel B (category-weighted):** Per-category HHI (HS 8541 and HS 8542 separately), combined via the value-weighted formula (Equation (3)).

Cross-channel aggregation. Arithmetic mean (Equation (2)).

Fallback basis. `BOTH` when aggregate and per-category data are available (the normal case for all EU-27 states). `A_ONLY` / `B_ONLY` / `NONE` as for Axis 1.

Boundary conditions. Countries with zero recorded semiconductor imports receive axis score null.

4.4 Axis 4: Defence Industrial Dependency

Concept. Concentration of a country’s conventional arms imports across supplier countries.

Source. SIPRI Arms Transfers Database (ATD). Unit of measurement: SIPRI Trend Indicator Values (TIV), which provide a common metric for comparing transfers of different weapons systems.

Temporal window. Defence transfers are lumpy. To mitigate year-to-year volatility, a **six-year** rolling window is applied: share vectors are computed from the sum of TIV flows over years 2019–2024 for v 1.0. The six-year aggregation window is a fixed parameter of the v 1.0 methodology definition. Shortening or extending the window constitutes a major-version change under the governance protocol (Section 9).

Channel decomposition.

- **Channel A (aggregate):** HHI computed over partner-country shares of total TIV imports across the six-year window.
- **Channel B (capability-block weighted):** Per-capability-block HHI (e.g., combat aircraft, armoured vehicles, naval vessels, missiles, air-defence systems), combined via the value-weighted formula (Equation (3)). Capability blocks follow the SIPRI category taxonomy.

Cross-channel aggregation. Arithmetic mean (Equation (2)).

Warning codes. W-DEF (sparse transfer record: raised when country i records fewer than three transfer events in the six-year window, yielding an HHI that may reflect sampling noise rather than structural dependence).

Boundary conditions. Countries with zero recorded arms imports on both channels across the six-year window receive axis score null and are subject to the missing-axis coverage rule (Section 5.2). In the v 1.0 snapshot, every EU-27 member state records at least one transfer event in the 2019–2024 window; this boundary condition is therefore not triggered for any country in the published output.

4.5 Axis 5: Critical Inputs / Raw Materials

Concept. Concentration of a country’s imports of critical raw materials across partner countries.

Source. Eurostat Comext at CN8 level. The material scope covers a curated list of critical raw materials aligned with the EU Critical Raw Materials Act (e.g., rare earths, lithium, cobalt, manganese, tungsten, gallium, germanium, titanium, and platinum-group metals).

Channel decomposition.

- **Channel A (aggregate):** HHI computed over partner-country shares of the combined import value of all critical materials.
- **Channel B (material-group weighted):** Per-material-group HHI, combined via the value-weighted formula (Equation (3)). Material groups are defined in the methodology registry.

Cross-channel aggregation. Arithmetic mean (Equation (2)).

Fallback basis. As for Axis 1 (BOTH / A_ONLY / B_ONLY / NONE).

Boundary conditions. Countries with zero recorded critical-material imports receive axis score null.

4.6 Axis 6: Logistics and Freight Dependency

Concept. Concentration of a country’s freight transport linkages across partner countries and transport modes.

Source. Eurostat transport statistics. Scope: maritime, road, rail, and air freight flows (tonnage).

Channel decomposition.

- **Channel A (mode HHI):** For each transport mode, the HHI is computed over partner-country shares of tonnage. The mode-level scores are then combined as a tonnage-weighted average across modes.
- **Channel B (partner HHI per mode):** For each mode, the partner-level HHI is computed and tonnage-weighted across modes, analogous to the category-weighted variant (Equation (3)).

Cross-channel aggregation. Arithmetic mean (Equation (2)).

Boundary conditions. Landlocked countries (AT, CZ, HU, LU, SK) have no maritime freight data; the maritime mode is excluded from their tonnage-weighted average (denominator adjusts automatically). Countries with complete absence of transport data receive axis score null.

4.7 Axis Summary

Table 3 provides a compact reference.

Table 3. Axis registry summary.

Axis	Domain	Ch. A	Ch. B	Aggregation variant
1	Financial sovereignty	BIS inward claims	IMF CPIS debt	Arithmetic mean
2	Energy dependency	Per-fuel HHI (no channels)		Fuel average
3	Technology / semiconductor	Aggregate HHI	Category-wtd	Arithmetic mean
4	Defence industrial	Aggregate HHI	Capability-wtd	Arithmetic mean (6-yr)
5	Critical inputs	Aggregate HHI	Material-wtd	Arithmetic mean
6	Logistics / freight	Mode HHI	Partner HHI	Tonnage-weighted

5 Aggregation and Classification

5.1 Aggregation Pipeline

The end-to-end computation from raw data to published output follows six sequential stages:

1. **Share-vector construction.** For each country–axis pair (and, where applicable, each channel or fuel category), partner-level flows are normalised to sum-to-one share vectors.

2. **Concentration scoring.** The concentration formula (Equation (1)) is applied to each share vector, yielding channel-level, fuel-level, or mode-level scores as appropriate for the axis (see Section 4).
3. **Axis-level aggregation.** Channel scores are combined via the arithmetic mean (Equation (2)), fuel scores via the fuel average (Equation (4)), or tonnage-weighted scores via the category-weighted formula (Equation (3)), according to the axis-specific variant.
4. **Composite averaging.** The six axis scores are arithmetically averaged to produce the composite ISI_i (Equation (5)).
5. **Rounding.** All published scores—axis-level and composite—are rounded to eight decimal places (`ROUND_PRECISION = 8`, round-half-to-even). Rounding is applied before classification (step 6) and before hashing (Section 7.2). Classification operates exclusively on rounded values.
6. **Classification.** The rounded composite is mapped to one of four ordinal tiers (Table 2) by a single classification function in the classification module.

No step in this pipeline involves discretionary judgement. Given identical input data, the output is fully deterministic.

5.2 Missing-Axis Protocol

If axis a is non-computable for country i (due to null channels, zero flows, or data-coverage failure), the composite is computed over the remaining axes:

$$ISI_i = \frac{1}{|\mathcal{A}_i|} \sum_{a \in \mathcal{A}_i} M_i^{(a)} \quad (6)$$

where $\mathcal{A}_i \subseteq \{1, \dots, 6\}$ is the set of computable axes for country i . If $|\mathcal{A}_i| < 4$, the country is excluded from the ranked output and flagged `W-COV`.

v 1.0 status. In the `ISI v 1.0` snapshot (EU-27, vintage 2024), all six axes are computable for all 27 countries. The denominator in Equation (5) is therefore fixed at 6 for every country in the published output. The fallback rule defined above is not triggered in v 1.0 but remains part of the versioned methodology specification.

5.3 Ranking Methodology

Countries are ranked in descending order of the composite score. Ties are broken lexicographically by ISO 3166-1 alpha-2 code. The ranking is ordinal; no cardinality is attached to rank differences.

5.4 Vintage and Periodicity

Each published snapshot carries a vintage year t . The vintage refers to the most recent data year for which the majority of axes have coverage. Due to source-specific publication lags (see

Section 2.3), a snapshot published in calendar year $t + 1$ or $t + 2$ may carry vintage year t . For v 1.0, the vintage year is **2024**.

The vintage year is recorded in the snapshot metadata and in the integrity chain (Section 7).

6 Scenario Engine

6.1 Purpose

The scenario engine provides a first-order sensitivity capability. It answers questions of the form: “If country i ’s score on axis a were to shift by a given percentage, how would the composite score and rank change?”

The engine operates **at the axis-score level**. It does not decompose shocks to individual partners or sub-categories. It is a mechanical sensitivity tool, not a forecasting model.

6.2 Multiplicative Adjustment Model

A scenario is defined by a set of axis-level adjustments for a single country i . For each axis a , let $\alpha_a \in [-0.20, +0.20]$ be the proportional adjustment. The simulated axis score is:

$$\widetilde{M}_i^{(a)} = \text{clamp}(M_i^{(a)} \times (1 + \alpha_a), 0, 1) \quad (7)$$

where

$$\text{clamp}(x, a, b) = \min(\max(x, a), b)$$

The adjustment bounds $[-0.20, +0.20]$ are hard-coded in the backend. They constrain scenarios to plausible first-order perturbations and prevent the generation of extreme or nonsensical counterfactuals.

6.3 Composite Recomputation

After adjustments are applied, the simulated composite is computed via the standard methodology:

$$\widetilde{\text{ISI}}_i = \frac{1}{6} \sum_{a=1}^6 \widetilde{M}_i^{(a)}$$

The simulated composite is rounded to eight decimal places and classified using the same thresholds (Table 2) and the same function (`methodology.compute_composite`) as the baseline computation.

6.4 Rank Recomputation

The simulated country score is inserted into the baseline ranking of all $N = 27$ countries. Other countries’ scores remain at their baseline values. The resulting rank change quantifies the structural sensitivity of country i ’s position to the specified axis-level perturbation.

6.5 Constraints

1. **Unit-interval bounds.** No simulated axis score may fall below 0 or exceed 1. The clamping function enforces this.
2. **Adjustment bounds.** Each α_a is restricted to $[-0.20, +0.20]$. Inputs outside this range are rejected by the backend before computation.
3. **Single-country scope.** A scenario adjusts axis scores for one country at a time. Cross-country or general-equilibrium scenarios are not supported.

6.6 Use Cases

Typical applications include:

- *Axis sensitivity.* Setting $\alpha_a = -0.20$ for a single axis reveals which axes most influence a country's composite rank.
- *Diversification benefit.* Reducing a high-concentration axis by $\alpha_a = -0.10$ quantifies the composite improvement.
- *Stress test.* Increasing all axes by $\alpha_a = +0.20$ simultaneously shows the worst-case composite under the permitted perturbation envelope.

6.7 Limitations of the Scenario Engine

The engine models axis-score perturbations, not real economic adjustments. It does not capture:

- Partner-level reallocation (which partners gain or lose share).
- Second-round effects or substitution elasticities.
- Correlation between axes (e.g., an energy shock that simultaneously affects logistics costs).
- Time dynamics—all scenarios are instantaneous, single-period counterfactuals.

6.8 Isolation Guarantee

The scenario engine is a read-only branch of the computation pipeline (Section 15.5). The following normative constraints SHALL hold for every scenario invocation:

1. Scenario execution SHALL NOT modify any file in the snapshot directory.
2. Scenario execution SHALL NOT write to the methodology registry.
3. Scenario execution SHALL NOT produce, modify, or invalidate country hashes, snapshot hashes, or signatures.
4. Scenario outputs SHALL NOT be signed.
5. Scenario results are ephemeral: they exist only in the API response body and are not persisted in any durable store visible to the canonical pipeline.
6. Scenario results are outside the integrity chain. No hash digest or signature covers scenario-derived scores.

Consumers of scenario data **MUST** treat it as unsigned and non-authoritative.

Users should treat scenario outputs as indicative sensitivity measures, not as predictions.

7 Integrity Chain

7.1 Design Rationale

Published index scores are only credible if consumers can verify that the scores have not been altered after computation. The ISI embeds a cryptographic integrity chain in every published snapshot. The chain provides three guarantees:

1. **Tamper evidence.** Any post-publication modification to a country score, axis score, weight, threshold, or classification will invalidate the hash chain.
2. **Authorship attestation.** The Ed25519 signature over the snapshot hash binds the publication to a specific signing key, allowing consumers to verify that the snapshot was produced by the authorised computation environment.
3. **Offline verification.** Verification requires only the published snapshot files, the public key, and a standard cryptographic library. No access to the computation environment or raw data is needed.

7.2 Country-Level Hash

For each country i in vintage year t , a canonical string representation is constructed by concatenating the following fields in fixed order:

1. ISO 3166-1 alpha-2 country code.
2. Vintage year.
3. Methodology version string (e.g., 1.0).
4. Six axis scores $M_i^{(1)}, \dots, M_i^{(6)}$, each represented as a fixed-point decimal with eight digits after the decimal point. Null axes are encoded as the string `null`.
5. Composite score (eight-decimal fixed-point representation).
6. Weight vector (the axis weights used; $\frac{1}{6}$ for all six axes in v 1.0).
7. Classification thresholds (the four boundary values).
8. Fallback-basis vector (one entry per axis: BOTH, A_ONLY, B_ONLY, or NONE).

The country-level hash is:

$$h_i = \text{SHA-256}(\text{canonical}_i)$$

The exact serialisation format (field separators, encoding) is defined in the integrity serialisation module of the backend and is frozen under major version 1.x. Any change to the serialisation order or encoding constitutes a major-version increment.

7.3 Snapshot Hash

The snapshot hash aggregates all country-level hashes into a single digest. Country hashes are sorted in ascending lexicographic order of country code and concatenated:

$$H_t = \text{SHA-256}(h_{\text{AT}} \parallel h_{\text{BE}} \parallel \dots \parallel h_{\text{SK}})$$

where \parallel denotes concatenation.

7.4 Digital Signature

The snapshot hash is signed using Ed25519 [4]:

$$\sigma_t = \text{Ed25519-Sign}(sk, H_t)$$

where sk is the signing private key held by the publication authority. The corresponding public key pk is distributed via the project's public key registry and pinned in the snapshot metadata.

7.5 Published Artefacts

Each snapshot publication includes three integrity files:

MANIFEST.json

Lists every output file in the snapshot (score files, metadata, methodology registry) with its SHA-256 hash. Consumers can verify file-level integrity by recomputing hashes and comparing.

HASH_SUMMARY.json

Contains the per-country hashes h_i and the aggregate snapshot hash H_t . This file enables country-level verification without recomputing the full pipeline.

SIGNATURE.json

Contains the Ed25519 signature σ_t over H_t and the public key identifier used for signing.

7.6 Verification Procedure

A consumer verifying snapshot t performs the following steps:

1. Recompute each country hash h_i from the published score record using the canonical serialisation defined in the integrity serialisation module.
2. Concatenate country hashes in lexicographic order and compute H'_t .
3. Verify that $H'_t = H_t$ (the published snapshot hash in `HASH_SUMMARY.json`).
4. Verify the Ed25519 signature: $\text{Ed25519-Verify}(pk, H_t, \sigma_t) = \text{true}$ using the public key referenced in `SIGNATURE.json`.

If any step fails, the snapshot is considered compromised.

7.7 Key Management

The signing key pair is generated once per major version of the index. Key rotation occurs at major-version boundaries (see Section 9). The public key is published in the project repository and in `SIGNATURE.json`. The private key is stored in an isolated environment and is never exposed to the general computation pipeline.

8 Limitations

The ISI is subject to the following structural limitations. These are inherent to the methodology and cannot be resolved by parameter tuning, software correction, or improved data quality within the v 1.0 framework.

- L1. Re-export and entrepôt masking.** The index attributes flows to the immediate bilateral partner, not to the country of origin or ultimate destination. Entrepôt economies (notably the Netherlands and Belgium within the EU-27) inflate their share in partner-level distributions for goods that merely transit through their territory. This affects axes 3 (semiconductors), 5 (critical raw materials), and 6 (logistics) most acutely. No harmonised re-export adjustment exists across all source datasets; consequently, the ISI does not strip out re-export flows.
- L2. Small-economy amplification.** Countries with very small absolute import volumes (e.g., CY, MT, LU) tend to exhibit high HHI scores because a single large shipment from one partner can dominate the share distribution. The resulting concentration score is mechanically correct but may overstate structural dependence: a small economy can often switch suppliers at lower absolute cost than a large economy with the same score.
- L3. Geographic determinism (Axis 6).** Freight flows are partly determined by geography. Landlocked countries (AT, CZ, HU, LU, SK) necessarily concentrate road and rail freight on neighbouring partners, producing high logistics HHI values that reflect physical proximity rather than strategic vulnerability. The index does not adjust for geographic proximity or contiguity.
- L4. No domestic production offset.** The ISI measures concentration of *imports* only. It does not account for domestic production capacity. A country that imports a concentrated bundle of semiconductors but also manufactures a large share domestically receives the same axis score as a country with identical import patterns but no domestic production. Import concentration is therefore a necessary but not sufficient indicator of actual supply vulnerability.
- L5. Equal weighting arbitrariness.** The composite assigns equal weight to all six axes. This is transparent and neutral but may not reflect the relative strategic importance of each domain for a specific country. A small island state's energy-import concentration may be existentially significant, while its defence-transfer concentration may be economically negligible. The composite does not capture this asymmetry. Users requiring domain-specific analysis should examine axis-level scores directly.

- L6. Intra-EU trade inclusion.** Bilateral flows between EU member states are included in the share vectors. A country whose imports are concentrated on a single EU partner (e.g., Germany) receives a high HHI, even though the flow occurs within the Single Market and is subject to EU regulatory safeguards. This may overstate effective external dependence for countries whose concentration is predominantly intra-EU. An alternative specification—excluding intra-EU flows—would require a separate methodology variant and is reserved for future consideration.
- L7. Single vintage; no time-series comparability.** Only ISI 2024 is materialisable under v 1.0 (Section 2.3). Cross-temporal comparisons are not supported. Users cannot infer trends, convergence, or divergence from a single snapshot. Time-series extension requires additional source vintages and constitutes a future methodology release.
- L8. No formal verification of the pipeline.** The computation pipeline is empirically validated through the test suite (Section 13) but has not been subjected to formal verification (proof-carrying code, model checking, or certified compilation). The determinism invariants (Section 10.2) are contractual obligations enforced by testing, not by mathematical proof. Residual implementation defects that escape the test suite remain possible.

These limitations are permanent features of v 1.0. They are documented here to enable informed use of the index and to prevent over-interpretation of the published scores.

9 Governance and Versioning

9.1 Custodian

The ISI methodology is maintained by the **International Sovereignty Institute — Research Unit**. All methodology changes, snapshot publications, and key-management decisions are the responsibility of the custodian.

9.2 Versioning Scheme

The ISI follows a three-part version number: `major.minor.patch`. The current version is **v 1.0.0**.

Major version increment. A major version increment (e.g., $1.x.x \rightarrow 2.0.0$) is REQUIRED if any of the following changes is made:

- The aggregation rule (Equation (5)) changes.
- The number of axes changes (axes added or removed).
- Any classification threshold value (Table 2) changes.
- The rounding precision changes (currently 8 decimal places).
- The hashing algorithm changes (currently SHA-256).
- The signature algorithm changes (currently Ed25519).

- The canonical serialisation format (Section 10.4) changes.
- The hash-input field order or delimiter changes.
- The concentration formula (Equation (1)) changes.
- The defence temporal window changes.
- The axis-weighting scheme changes.

A major-version change resets the minor and patch counters to 0 and requires a new methodology document.

Minor version increment. A minor version increment (e.g., 1.0.x → 1.1.0) is REQUIRED if any of the following changes is made, provided no major-version trigger applies:

- Axis description text or metadata is revised without affecting the mathematical definition.
- The warning-code taxonomy (Section C) is expanded.
- The snapshot metadata schema (Section 11.5) is extended with new optional fields that do not enter the hash preimage.
- Country coverage is extended or reduced.
- A data-ingestion bug is corrected, producing different scores for one or more countries.
- A supplementary data source is added or replaced.

A minor-version change resets the patch counter to 0.

Patch version increment. A patch version increment (e.g., 1.0.0 → 1.0.1) is permitted only if *all* of the following conditions hold:

- The change is limited to documentation corrections, typographic fixes, or non-normative commentary.
- No materialised output (score files, hash digests, signatures) is affected.
- The `methodology_registry.json` semantics are unchanged.
- No snapshot is recomputed.

Patch versions do not alter the computation and therefore do not require a new snapshot.

Version-definition immutability. Once a version number has been assigned to a published snapshot, the association between that version number and the corresponding methodology parameters SHALL NOT be altered. The methodology registry SHALL be append-only: existing entries SHALL NOT be mutated or removed. This rule applies to all three version components (major, minor, patch).

9.3 Change Protocol

Proposed methodology changes follow a four-stage protocol:

1. **Proposal.** A written proposal specifies the change, its rationale, its impact on published scores (if any), and the affected version component (major or minor).
2. **Impact assessment.** The change is applied retroactively to the most recent two vintage years. The resulting score differences are tabulated and reviewed.
3. **Review.** The proposal and impact assessment are reviewed by at least one independent methodologist not involved in the original proposal.
4. **Publication.** If approved, the change is incorporated into the next version release. The methodology document, changelog, and version number are updated simultaneously. The previous version remains archived and accessible.

9.4 Data-Revision Policy

Statistical agencies routinely revise historical data. When a source revision affects the input data for a published vintage, the following rules apply:

- **No retroactive recalculation.** Published snapshots are not recomputed. The integrity chain binds each snapshot to the data as it existed at computation time.
- **Prospective incorporation.** Revised data are incorporated into the next vintage computation. The changelog records which source revisions were absorbed.
- **Erratum.** If a source revision reveals a material error in a published snapshot (defined as a composite-score change exceeding 0.01 for any country), an erratum is published identifying the affected countries and scores. The original snapshot and its integrity chain remain unchanged.

9.5 Publication Schedule

Snapshots are published annually. The target publication window is Q2 of the calendar year, reflecting the typical lag in source-data availability. The precise date is announced in the project repository at least 30 days in advance.

9.6 Archival

All published snapshots, methodology documents, changelogs, public keys, and integrity records are archived in a public repository with persistent identifiers. No published artefact may be deleted or overwritten. Corrections are published as addenda, not as replacements.

10 Determinism and Computational Guarantees

This section formalises the determinism properties of the ISI computation pipeline. Every guarantee stated here is a binding contract: a conforming implementation *must* satisfy all invariants, and a failure to satisfy any single invariant constitutes a non-conforming build.

10.1 Scope of Determinism

The ISI pipeline is **fully deterministic**: given identical input data, identical methodology-version parameters, and a conforming arithmetic environment, the pipeline produces bit-identical output across invocations, platforms, and calendar time. Determinism is not a best-effort aspiration; it is an architectural invariant enforced by the design constraints enumerated below.

Determinism extends to all published artefacts: axis scores, composite scores, classifications, rankings, warning codes, hash digests, and the snapshot signature. It does *not* extend to execution time, memory consumption, or log output, which are platform-dependent observables.

10.2 Formal Invariants

The following invariants hold for every conforming implementation of the ISI v 1.0 pipeline. Each invariant is designated **D-*n*** and is referenced in the test-suite specification (Section 13).

D-1 Input–output identity. Let \mathcal{D} be a fixed input dataset and θ a fixed methodology-parameter vector. For any two invocations r_1, r_2 of the pipeline on (\mathcal{D}, θ) :

$$\text{output}(r_1) = \text{output}(r_2)$$

where equality is byte-level identity of the serialised output.

D-2 Platform invariance. Let P_1, P_2 be any two platforms satisfying the arithmetic assumptions of Section 10.3. Then:

$$\text{output}(P_1, \mathcal{D}, \theta) = \text{output}(P_2, \mathcal{D}, \theta)$$

D-3 Time invariance. The pipeline output depends only on (\mathcal{D}, θ) . No timestamp, random seed, environment variable, locale setting, or system clock reading enters the computation. Formally: for any calendar times t_1, t_2 :

$$\text{output}(t_1, \mathcal{D}, \theta) = \text{output}(t_2, \mathcal{D}, \theta)$$

D-4 Order invariance of input rows. The pipeline sorts all input records into a canonical order before computation. Permuting the row order of the input dataset produces identical output:

$$\forall \pi \in \mathcal{S}_{|\mathcal{D}|} : \text{output}(\pi(\mathcal{D}), \theta) = \text{output}(\mathcal{D}, \theta)$$

D-5 Rounding closure. All published scores are rounded to eight decimal places (round-half-to-even) *before* any downstream consumption (classification, hashing, serialisation). Formally: let $R_8(x)$ denote the round-half-to-even function at precision 8. Then for every published score s :

$$R_8(s) = s$$

That is, every published score is a fixed point of the rounding operator.

D-6 Hash determinism. The country-level hash (Section 7.2) and snapshot hash (Section 7.3) are fully determined by the published scores and the canonical serialisation format. No non-deterministic input (timestamp, hostname, process identifier) enters the hash preimage.

D-7 Classification determinism. Classification is a pure function of the rounded composite score and the threshold vector. For a given rounded composite s and threshold vector $\tau = (0.15, 0.25, 0.50)$:

$$\text{classify}(s, \tau) = \begin{cases} \text{highly_concentrated} & \text{if } s \geq 0.50 \\ \text{moderately_concentrated} & \text{if } 0.25 \leq s < 0.50 \\ \text{mildly_concentrated} & \text{if } 0.15 \leq s < 0.25 \\ \text{unconcentrated} & \text{if } s < 0.15 \end{cases}$$

No stochastic or environment-dependent branching is permitted.

D-8 Ranking determinism. The ranking function is a deterministic total order. The primary sort key is the rounded composite score (descending). The secondary sort key (tie-breaker) is the ISO 3166-1 alpha-2 code (ascending, lexicographic). This total order is unique for any distinct set of (score, code) pairs.

D-9 Scenario determinism. The scenario engine (Section 6) is a deterministic function of the baseline scores and the adjustment vector $(\alpha_1, \dots, \alpha_6)$. Clamping (Equation (7)) is deterministic; no randomised rounding or stochastic perturbation is applied.

D-10 Missing-axis determinism. The missing-axis protocol (Section 5.2) is a deterministic function of the set of computable axes \mathcal{A}_i . The composite denominator is $|\mathcal{A}_i|$, not a stochastic or heuristic estimate.

10.3 Floating-Point Assumptions

The pipeline assumes **IEEE 754 binary64** (double-precision) floating-point arithmetic with the following constraints:

1. **Rounding mode.** The default IEEE 754 rounding mode (*round-to-nearest, ties-to-even*) must be active throughout the computation. No midstream rounding-mode changes are permitted.

2. **Operation order.** Summations over partner shares and over axis scores follow a fixed left-to-right accumulation order after sorting the operands into canonical sequence (partner code ascending for share vectors; axis index ascending for composite averaging). The accumulation order is specified to eliminate platform-dependent reordering that could produce different floating-point round-off patterns.
3. **No fused multiply-add.** Fused multiply-add (FMA) instructions may produce different round-off behaviour from separate multiply-then-add sequences. Conforming implementations must disable FMA contraction or demonstrate bit-identical results under FMA.
4. **Decimal rounding.** The final rounding step (`ROUND_PRECISION = 8`, round-half-to-even) is performed using Python's `decimal.Decimal` module or an equivalent arbitrary-precision decimal library, not by native binary64 rounding. This eliminates the double-rounding problem inherent in converting binary64 to decimal.
5. **No extended precision.** Intermediate calculations must not use x87 80-bit extended precision. All intermediates are stored and operated upon as binary64. On x86 platforms, the compiler must be configured to use SSE2 instructions (not x87 FPU) for floating-point arithmetic, or the FPU control word must be set to 64-bit precision.

These constraints are sufficient to guarantee **D-1** and **D-2** across all platforms that conform to IEEE 754-2008.

10.4 Canonical Serialisation

Deterministic output requires a deterministic serialisation format. The `ISI` pipeline serialises all published data according to the following normative rules. Violation of any rule produces a non-conforming snapshot whose hash **MUST** differ from that of a conforming snapshot.

1. **Character encoding.** All output files **SHALL** use UTF-8 without byte-order mark (BOM). No other encoding is permitted.
2. **Newline convention.** Line endings **SHALL** be Unix-style (`\n`, U+000A). No carriage-return characters (`\r`, U+000D) **SHALL** appear in any output file.
3. **No trailing whitespace.** No line in any output file **SHALL** end with a space (U+0020) or horizontal tab (U+0009) character before the line-feed terminator.
4. **JSON formatting.** JSON output files **SHALL** use two-space indentation, no trailing commas, and keys sorted in *ascending lexicographic order* (byte-wise comparison of UTF-8 code units) at every nesting level. Floating-point values **SHALL** be serialised as fixed-point decimal strings with exactly eight digits after the decimal point and a period (U+002E) as the decimal separator (e.g., `"0.34218753"`, not `0.34218753` or `"3.4218753e-01"`). No locale-dependent thousands separator, comma-as-decimal, or scientific notation is permitted.
5. **CSV formatting.** CSV output files **SHALL** use comma delimiters, double-quote text-field quoting, and header rows matching the schema defined in the output specification

(Section 11.3). Fields SHALL be ordered by the fixed column schema; no column reordering is permitted across versions.

6. **Locale independence.** No serialisation step SHALL read or depend on the system locale, the `LC_NUMERIC` environment variable, or any locale-sensitive formatting library. All numeric formatting SHALL use the explicit rules defined in items 4 and 5.
7. **Stable dictionary ordering.** Where an output structure is constructed from an in-memory dictionary or hash map, the serialisation step SHALL sort keys into ascending lexicographic order before writing. Reliance on insertion-order preservation (e.g., Python 3.7+ dictionary ordering) is insufficient; an explicit `sorted()` call is REQUIRED.
8. **Deterministic country-code sorting.** Country codes in all output files (CSV rows, JSON arrays, hash-input concatenation) SHALL be sorted in ascending lexicographic order of the two-character ISO 3166-1 alpha-2 code.
9. **Hash-input strings.** The canonical string used as the SHA-256 preimage for country-level hashing (Section 7.2) concatenates fields using the pipe character (`|`) as a delimiter, in the fixed order specified in the integrity-chain section. No whitespace padding is applied. Null axis scores are encoded as the literal string `null`.

Hash-breakage clause. If any of the serialisation rules above is violated—even if the underlying numeric values are correct—the resulting SHA-256 hash SHALL differ from the hash of a conforming serialisation. This is a design feature, not a defect: it ensures that serialisation conformance is verifiable through the integrity chain.

10.5 Prohibited Operations

The following operations are prohibited in the computation pipeline:

- **Random number generation.** No call to a pseudo-random or true-random number generator is permitted at any stage.
- **System clock queries.** No timestamp or system clock reading may influence a computation step. Timestamps in metadata (e.g., the `computed_at` field in the snapshot manifest) are recorded as post-computation annotations and are excluded from the hash preimage.
- **Environment-dependent branching.** No control-flow decision may depend on the host-name, operating system, CPU architecture, locale, timezone, or any environment variable.
- **Unordered iteration.** No computation may iterate over a hash-map, set, or dictionary without first imposing a canonical sort order. In Python, this implies explicit `sorted()` calls on dictionary keys wherever iteration order affects the result.
- **Floating-point equality comparison for branching.** Classification uses strict inequality comparisons against decimal threshold constants. No floating-point equality test (e.g., `== 0.25`) is used in any branching condition, except where both operands are known to be exactly representable in binary64.

10.6 Hash Dependency Graph

The integrity chain imposes a strict dependency order on computed values. No hash may be computed until all of its upstream inputs have been finalised (rounded and serialised). The dependency graph is:

- Stage 1. Channel scores.** Per-partner share vectors \rightarrow channel-level HHI (Equation (1)).
- Stage 2. Axis scores.** Channel scores \rightarrow axis-level aggregation (Equations (2) to (4)).
- Stage 3. Composite score.** Six axis scores \rightarrow arithmetic mean (Equation (5)).
- Stage 4. Rounding.** All scores rounded to 8 decimal places (round-half-to-even).
- Stage 5. Classification.** Rounded composite \rightarrow four-tier label (Table 2).
- Stage 6. Country hash.** Canonical serialisation of rounded scores, classification, and metadata $\rightarrow h_i = \text{SHA-256}(\text{canonical}_i)$ (Section 7.2).
- Stage 7. Snapshot hash.** Sorted country hashes $\rightarrow H_t = \text{SHA-256}(h_{\text{AT}} \parallel \dots \parallel h_{\text{SK}})$ (Section 7.3).
- Stage 8. Signature.** Snapshot hash $\rightarrow \sigma_t = \text{Ed25519-Sign}(sk, H_t)$ (Section 7.4).

A stage may commence only after all inputs from the preceding stage are finalised. Rounding (Stage 4) is the single irreversible precision-reduction step; all subsequent stages (5–8) consume only rounded values. Reversing the order of Stages 4 and 5 (i.e., classifying before rounding) would violate invariant **D-5** and produce inconsistent hash preimages.

10.7 Determinism Certification

A conforming implementation is certified deterministic if it passes the full test suite defined in Section 13, including:

1. **Golden-file tests.** The output for a fixed reference dataset matches a pre-computed golden file byte-for-byte.
2. **Cross-platform tests.** The same golden-file comparison passes on at least two distinct platforms (e.g., x86-64 Linux and ARM64 macOS).
3. **Idempotency tests.** Running the pipeline twice on the same input produces byte-identical output (verifying **D-1**).
4. **Permutation tests.** Randomly permuting the input-row order produces identical output (verifying **D-4**).

11 Snapshot Materialisation and Registry Architecture

This section specifies the file-system structure, registry schema, and immutability contract that govern a published ISI snapshot. The architecture is designed so that every snapshot is a self-contained, content-addressed, read-only artefact that can be verified without access to the computation environment.

11.1 Design Principles

Three principles govern the snapshot architecture:

- S-1. Self-containment.** A snapshot directory contains *all* information required to verify the integrity chain, reproduce the classification, and reconstruct the ranking. No external lookup, network request, or database query is required.
- S-2. Content addressing.** Every file in the snapshot is listed in the manifest with its SHA-256 hash. The manifest itself is covered by the snapshot signature. Content addressing ensures that any file substitution or corruption is detectable.
- S-3. Immutability.** Once published, no file within a snapshot directory may be modified, re-named, or deleted. Corrections are issued as separate addenda under the governance protocol (Section 9); they do not alter the original snapshot.

11.2 Directory Structure

A conforming snapshot for vintage year t and methodology version v is materialised in the following directory layout:

```
snapshots/v{v}/{t}/
  MANIFEST.json
  HASH_SUMMARY.json
  SIGNATURE.json
  methodology_registry.json
  scores/
    country_scores.csv
    axis_scores.csv
    channel_scores.csv
  metadata/
    snapshot_metadata.json
    warnings.csv
    fallback_basis.csv
```

Every file listed above is **mandatory**. No additional files may appear in a conforming snapshot directory. The path prefix `snapshots/` is a convention; the top-level directory name is not semantically significant.

11.3 Output Schema

11.3.1 Country Scores (`country_scores.csv`)

Each row corresponds to one country in the vintage. The fixed column schema is:

Table 4. Column schema for `country_scores.csv`.

Column	Type	Description
<code>country_code</code>	<code>str</code>	ISO 3166-1 alpha-2 code.
<code>vintage_year</code>	<code>int</code>	Vintage year t .
<code>methodology_version</code>	<code>str</code>	Version string (e.g., "1.0").
<code>axis_1_score</code>	<code>str</code>	Axis 1 score, 8-decimal fixed-point, or "null".
<code>axis_2_score</code>	<code>str</code>	Axis 2 score.
<code>axis_3_score</code>	<code>str</code>	Axis 3 score.
<code>axis_4_score</code>	<code>str</code>	Axis 4 score.
<code>axis_5_score</code>	<code>str</code>	Axis 5 score.
<code>axis_6_score</code>	<code>str</code>	Axis 6 score.
<code>composite_score</code>	<code>str</code>	Composite ISI score, 8-decimal fixed-point.
<code>classification</code>	<code>str</code>	One of the four classification labels.
<code>rank</code>	<code>int</code>	Ordinal rank (1 = most concentrated).
<code>computable_axes</code>	<code>int</code>	Number of computable axes $ \mathcal{A}_i $.

Scores are serialised as fixed-point decimal *strings* ("0.34218753"), not as floating-point numbers. This eliminates parser-dependent rounding artefacts in downstream tooling.

11.3.2 Axis Scores (`axis_scores.csv`)

One row per country–axis pair. Columns:

Table 5. Column schema for `axis_scores.csv`.

Column	Type	Description
<code>country_code</code>	<code>str</code>	ISO 3166-1 alpha-2 code.
<code>axis</code>	<code>int</code>	Axis index (1, ..., 6).
<code>score</code>	<code>str</code>	8-decimal fixed-point or "null".
<code>fallback_basis</code>	<code>str</code>	One of BOTH, A_ONLY, B_ONLY, NONE.
<code>warnings</code>	<code>str</code>	Comma-separated warning codes, or empty.

11.3.3 Channel Scores (`channel_scores.csv`)

One row per country–axis–channel triplet. This file provides the lowest-level computed scores for axes that use a two-channel decomposition. For Axis 2 (Energy), which uses fuel averaging, each fuel category is treated as a separate channel row.

Table 6. Column schema for `channel_scores.csv`.

Column	Type	Description
<code>country_code</code>	<code>str</code>	ISO 3166-1 alpha-2 code.
<code>axis</code>	<code>int</code>	Axis index.
<code>channel</code>	<code>str</code>	Channel identifier (A, B, or fuel/mode name).
<code>score</code>	<code>str</code>	8-decimal fixed-point or "null".
<code>partner_count</code>	<code>int</code>	Number of partners in the share vector.

11.4 Methodology Registry

Each snapshot includes a machine-readable methodology registry file (`methodology_registry.json`) that encodes every parameter required to reproduce the computation. The registry is a flat JSON object with the following mandatory keys:

Table 7. Methodology registry keys.

Key	Type	Description
<code>methodology_version</code>	<code>str</code>	"1.0"
<code>vintage_year</code>	<code>int</code>	2024
<code>country_set</code>	<code>str[]</code>	Sorted ISO alpha-2 codes
<code>axis_count</code>	<code>int</code>	6
<code>axis_weights</code>	<code>str[]</code>	Equal weights ($\frac{1}{6}$ each)
<code>round_precision</code>	<code>int</code>	8
<code>rounding_mode</code>	<code>str</code>	"half_even"
<code>classification_thresholds</code>	<code>obj</code>	Tier boundaries as decimal strings
<code>scenario_bounds</code>	<code>str[]</code>	["-0.20", "+0.20"]
<code>defence_window_years</code>	<code>int</code>	6
<code>min_computable_axes</code>	<code>int</code>	4
<code>hash_algorithm</code>	<code>str</code>	"SHA-256"
<code>signature_algorithm</code>	<code>str</code>	"Ed25519"
<code>energy_fuel_categories</code>	<code>str[]</code>	Gas, oil, solid fossil
<code>semiconductor_hs_codes</code>	<code>str[]</code>	HS 8541 and HS 8542

All numeric parameters are serialised as strings to preserve exact decimal representation. The registry file is included in the manifest hash and is therefore covered by the integrity chain.

11.5 Snapshot Metadata

The `snapshot_metadata.json` file records non-computational metadata:

Table 8. Snapshot metadata keys.

Key	Type	Description
<code>computed_at</code>	<code>str</code>	ISO 8601 timestamp of computation completion (not used in hashing).
<code>published_at</code>	<code>str</code>	ISO 8601 timestamp of publication (not used in hashing).
<code>pipeline_version</code>	<code>str</code>	Git commit hash of the computation codebase at build time.
<code>python_version</code>	<code>str</code>	Python interpreter version (e.g., "3.12.4").
<code>platform</code>	<code>str</code>	Platform identifier (e.g., "linux-x86_64").
<code>notes</code>	<code>str</code>	Free-text publication note (not used in hashing).

None of the metadata fields enter the hash preimage or affect the integrity chain. They are informational annotations only. The `computed_at` and `published_at` fields specifically violate time-invariance (**D-3**) if they were to enter the hash; their exclusion is deliberate.

11.6 Immutability Contract

Once a snapshot is published (i.e., the `published_at` field is set and the directory is committed to the public archive), the following constraints apply:

- I-1. No file modification.** No file within the snapshot directory may be modified after publication. This includes the manifest, score files, registry, metadata, and integrity files.
- I-2. No file deletion.** No file may be removed from the snapshot directory.
- I-3. No file addition.** No file may be added to the snapshot directory after publication. Supplementary materials (e.g., errata, addenda) are published as separate artefacts under a distinct path.
- I-4. No path renaming.** The directory path of a published snapshot may not be changed. Symbolic links to the snapshot are permitted but may not replace the original path.
- I-5. Addendum protocol.** Corrections to a published snapshot are published as addendum directories (`snapshots/v{v}/{t}/addenda/{seq}/`) containing the corrected files and a written justification. The original snapshot remains unmodified and its integrity chain remains valid.

Violation of any immutability constraint (**I-1** through **I-4**) renders the snapshot non-conforming and triggers a mandatory erratum under the governance protocol (Section 9.4).

Immutable Snapshot Law. The following normative statement summarises the immutability obligations and is intended to be citable by external auditors and consuming systems:

A published ISI snapshot is an append-only, content-addressed, cryptographically signed artefact. The snapshot directory SHALL NOT be modified, renamed, or deleted

after publication. The methodology registry SHALL be append-only: existing entries SHALL NOT be mutated or removed. Retroactive recomputation of a published snapshot is prohibited. Any material error discovered in a published snapshot SHALL be corrected by publishing a new version (v 1.1 or higher) under the governance protocol (Section 9.3); the original snapshot and its integrity chain SHALL remain intact and publicly accessible.

11.7 Manifest Construction

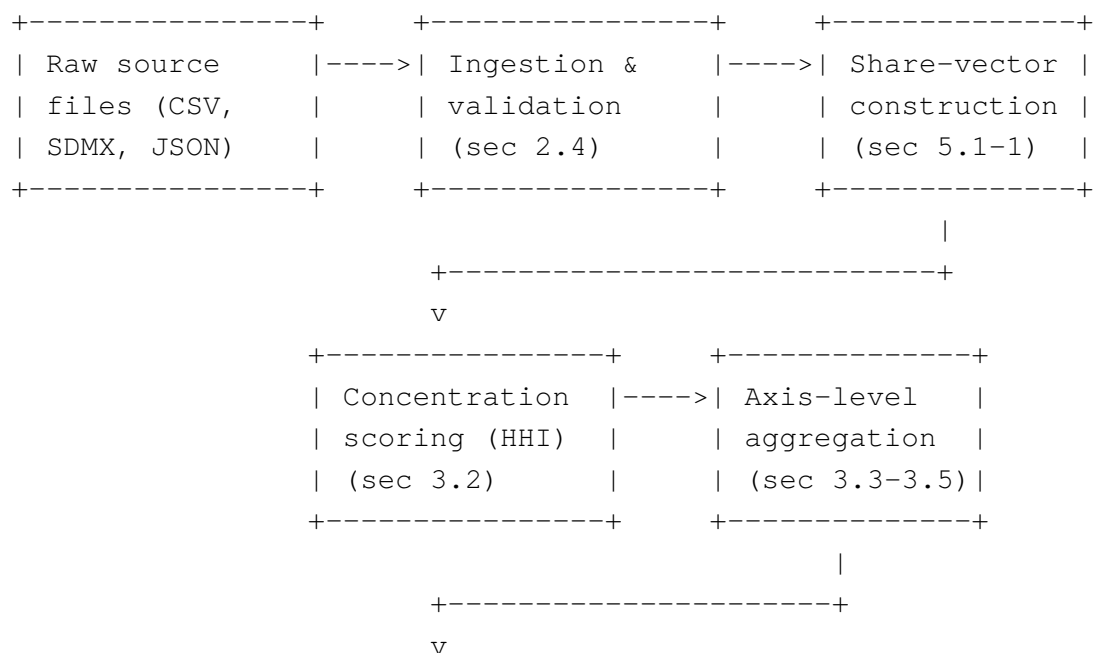
The manifest file (MANIFEST.json) is constructed as follows:

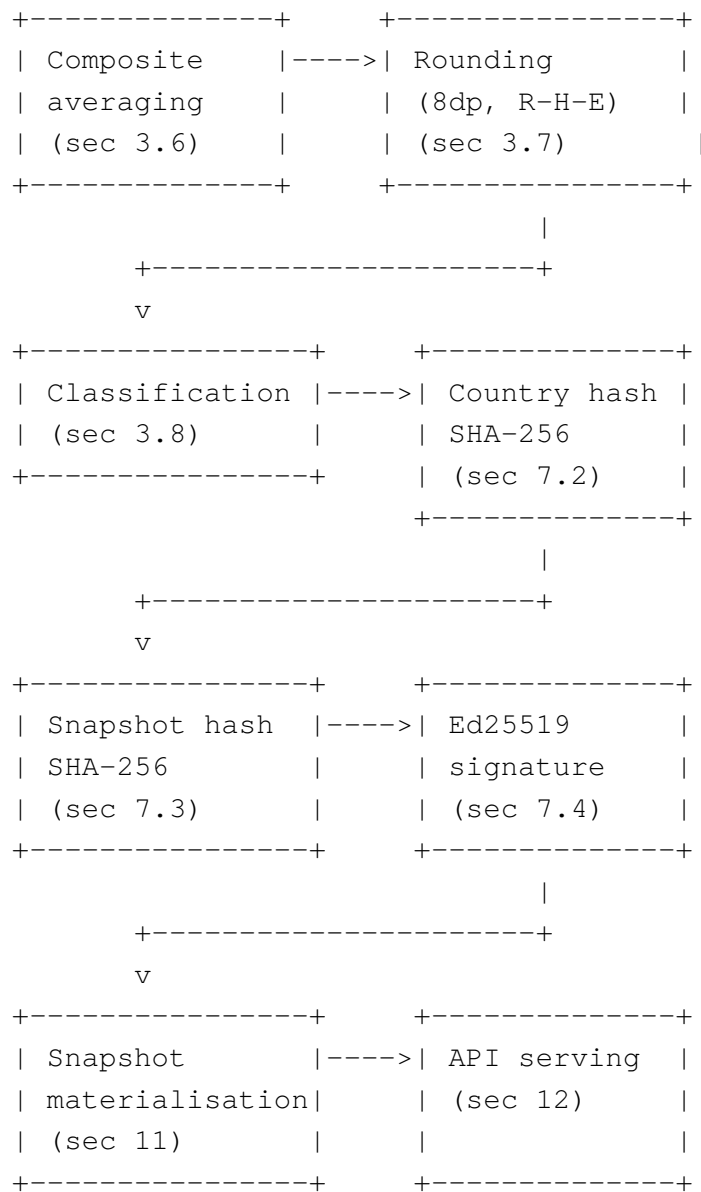
1. Enumerate all files in the snapshot directory excluding the three integrity files: MANIFEST.json, HASH_SUMMARY.json, and SIGNATURE.json.
2. Sort the file list lexicographically by relative path.
3. Compute the SHA-256 hash of each file (byte-level, no encoding transformation).
4. Serialise the result as a JSON object mapping relative file paths to hex-encoded SHA-256 hashes, with keys in sorted order.

The manifest is then itself covered by the snapshot hash (Section 7.3): the manifest hash is included in the HASH_SUMMARY.json as a separate field (manifest_hash), and the snapshot hash H_t is computed over the concatenation of all country hashes and the manifest hash.

11.8 Pipeline Architecture

The end-to-end data flow from raw sources to signed publication is illustrated below. Each arrow represents a deterministic, auditable transformation. No feedback loop or bidirectional dependency exists.





Every box in the diagram is a pure function of its inputs. No box reads from a mutable external state. The pipeline is executed exactly once per vintage; re-execution on identical inputs produces bit-identical output (**D-1**).

11.9 Hash Cascade Logic

The integrity chain forms a Merkle-like cascade with the following properties:

C-1 Leaf completeness.

Every country in the snapshot contributes exactly one leaf hash h_i . No country may be omitted from the cascade, even if its composite score is null (excluded countries receive a hash over a canonical null record).

C-2 Deterministic ordering.

Leaf hashes are concatenated in ascending lexicographic order of ISO 3166-1 alpha-2 country code. This ordering is invariant across platforms (D-2) and invocations (D-1).

C-3 Manifest binding.

The manifest hash is concatenated after the last country hash before computing H_t . This binds the file-level integrity check to the score-level integrity check in a single digest.

C-4 Single-root property.

The cascade terminates at a single root: the snapshot hash H_t . The Ed25519 signature covers only H_t ; individual country hashes are transitively authenticated through the cascade.

C-5 Tamper propagation.

Any modification to any score, classification, warning code, or metadata field in any country record changes h_i , which changes H_t , which invalidates σ_t . The propagation is guaranteed by the collision resistance of SHA-256.

11.10 Backward Compatibility of the Directory Schema

The directory schema defined above is frozen for major version 1.x. Specifically:

- No mandatory file may be removed.
- No mandatory column may be removed from a CSV schema.
- New *optional* columns may be appended to CSV files in minor versions, provided they are appended after all existing columns and do not alter the interpretation of existing columns.
- New *optional* keys may be added to JSON files in minor versions, provided they do not alter the semantics of existing keys.
- The `methodology_registry.json` key set may be extended but not contracted.

Any change that violates these rules constitutes a major-version increment under the governance protocol (Section 9).

12 API Contract and Serving Guarantees

This section specifies the programmatic interface through which consumers access published ISI data. The contract defines endpoint semantics, response formats, caching behaviour, error taxonomy, and backward-compatibility rules. The API is a **read-only** serving layer; it does not expose computation, ingestion, or administrative functions.

12.1 Transport and Protocol

- **Protocol.** HTTPS (TLS 1.2 or later). Plaintext HTTP requests receive a 301 Moved Permanently redirect to the HTTPS endpoint.
- **Content type.** All responses are served as JSON (`application/json`) with UTF-8 encoding unless otherwise specified per endpoint.

- **Authentication.** The public API is unauthenticated. Rate limiting is enforced per source IP address (see Section 12.5).
- **Base URL.** The canonical base URL is published in the project repository. All endpoint paths in this section are relative to the base URL.

12.2 Endpoint Taxonomy

The API exposes the following endpoints. All endpoints accept only GET requests. Other HTTP methods receive 405 Method Not Allowed.

12.2.1 GET /v1/snapshots

Returns the list of available snapshots.

Table 9. Response schema for GET /v1/snapshots.

Field	Type	Description
snapshots	obj[]	Array of snapshot summary objects.
.vintage_year	int	Vintage year.
.methodology_version	str	Version string.
.published_at	str	ISO 8601 publication timestamp.
.country_count	int	Number of countries in the snapshot.

12.2.2 GET /v1/snapshots/{vintage}/scores

Returns country-level scores for the specified vintage.

Path parameters. {vintage}: four-digit vintage year (e.g., 2024).

Query parameters.

country Optional. Comma-separated list of ISO 3166-1 alpha-2 codes. Filters the response to the specified countries.

format Optional. One of json (default) or csv. When csv, the response content type is text/csv; charset=utf-8 and the body contains the country_scores.csv content directly.

Response. The JSON response body is an object with a scores array whose elements follow the column schema of country_scores.csv (Table 4). All numeric scores are returned as strings (fixed-point, 8 decimal places).

12.2.3 GET /v1/snapshots/{vintage}/axes

Returns axis-level scores for all countries in the specified vintage. The response schema mirrors axis_scores.csv (Table 5).

12.2.4 GET /v1/snapshots/{vintage}/channels

Returns channel-level scores. The response schema mirrors `channel_scores.csv` (Table 6).

12.2.5 GET /v1/snapshots/{vintage}/integrity

Returns the integrity artefacts: country-level hashes, the snapshot hash, the manifest hash, and the Ed25519 signature. This endpoint provides sufficient information for a consumer to perform the full verification procedure (Section 7.6) without downloading the snapshot archive.

12.2.6 GET /v1/snapshots/{vintage}/registry

Returns the methodology registry (Section 11.4) as a JSON object.

12.2.7 GET /v1/scenario

Executes a scenario simulation and returns the results.

Query parameters.

vintage Required. Four-digit vintage year.

country Required. Single ISO 3166-1 alpha-2 code.

a1 ... a6 Optional. Axis-level adjustment factors, each a decimal in $[-0.20, +0.20]$. Omitted axes default to 0.

Response. The response contains the baseline scores, simulated scores, baseline rank, simulated rank, and the applied adjustment vector. All scores are returned as fixed-point strings.

12.3 ETag and Conditional Requests

Every response includes an `ETag` header whose value is the hex-encoded SHA-256 hash of the response body. Clients may issue conditional requests using `If-None-Match`; the server returns `304 Not Modified` if the `ETag` matches.

Because snapshot data is immutable (Section 11.6), `ETags` for snapshot endpoints are stable: the same request to the same vintage always produces the same `ETag`. This property is a direct consequence of determinism invariant **D-1** (Section 10.2).

Responses also include a `Cache-Control` header:

- Snapshot endpoints:
`Cache-Control: public, max-age=86400, immutable.`
- Scenario endpoint: `Cache-Control: public, max-age=3600.`
- Snapshot-list endpoint: `Cache-Control: public, max-age=3600.`

The `immutable` directive on snapshot endpoints signals to intermediary caches that the response will never change for the given URL, reflecting the immutability contract (Section 11.6).

12.4 Error Taxonomy

The API uses the following error response structure:

```
{"error": {"code": "<CODE>", "message": "<human-readable>"}}
```

Table 10 enumerates the error codes.

Table 10. API error codes.

HTTP	Code	Scope	Condition
400	INVALID_VINTAGE	Path	Vintage year is not a four-digit integer or is outside the range of published snapshots.
400	INVALID_COUNTRY	Query	Country code is not a valid ISO 3166-1 alpha-2 code.
400	INVALID_ADJUSTMENT	Query	Scenario adjustment factor is outside $[-0.20, +0.20]$ or is not a valid decimal.
400	MISSING_PARAMETER	Query	A required query parameter is absent.
404	VINTAGE_NOT_FOUND	Path	The requested vintage year does not correspond to a published snapshot.
404	COUNTRY_NOT_FOUND	Query	The requested country code is not present in the specified vintage.
405	METHOD_NOT_ALLOWED	Method	HTTP method other than GET was used.
429	RATE_LIMITED	Client	Client has exceeded the rate limit.
500	INTERNAL_ERROR	Server	An unexpected server-side error occurred. The response includes a correlation identifier for incident tracking.

12.5 Rate Limiting

Rate limiting is enforced per source IP address using a token-bucket algorithm. The default limits are:

Table 11. Default rate limits.

Endpoint group	Requests/minute	Burst
Snapshot data endpoints	60	10
Scenario endpoint	30	5
Snapshot list	60	10

When a client exceeds the rate limit, the server returns `429 Too Many Requests` with a `Retry-After` header indicating the number of seconds until the next permitted request.

12.6 Versioned Endpoint Prefix

All endpoints are prefixed with a version identifier (`/v1/`). The API version is *independent* of the methodology version: API `v1` may serve snapshots from methodology versions 1.0, 1.1, 1.2, etc.

A new API major version (`/v2/`) is introduced only when the response schema undergoes a backward-incompatible change. When a new API version is introduced, the previous version remains operational for a minimum of twelve months with a `Sunset` header indicating the deprecation date.

12.7 Backward-Compatibility Rules

Within a given API major version, the following changes are backward-compatible and do not require a version increment:

- Adding new optional query parameters.
- Adding new fields to JSON response objects.
- Adding new snapshot vintages.
- Adding new warning codes to the `warnings` field.

The following changes are **breaking** and require a new API major version:

- Removing or renaming an existing response field.
- Changing the type or semantics of an existing response field.
- Removing an endpoint.
- Changing the serialisation format of scores (e.g., from string to number).
- Altering the ETag computation algorithm.

12.8 Content Negotiation

Clients may request CSV output via the `format=csv` query parameter on score endpoints. The CSV output is byte-identical to the corresponding file in the snapshot archive, ensuring that consumers who download the snapshot and consumers who use the API receive the same data.

JSON is the default format. The `Accept` header is not used for content negotiation; the `format` query parameter takes precedence.

12.9 Health and Readiness

The API exposes two operational endpoints that are excluded from the versioned prefix:

GET `/healthz`

Returns `200 OK` with body `{"status": "ok"}` if the server process is running and able to accept connections. This endpoint performs no database or file-system checks. It is intended for load balancers and container orchestrators.

GET `/readyz`

Returns `200 OK` with body `{"status": "ready"}` if the server has loaded at least one snapshot into memory and is able to serve score requests. If no snapshot is loaded, it returns `503 Service Unavailable` with `{"status": "not_ready"}`.

Neither endpoint is subject to rate limiting or authentication. Neither endpoint returns cache headers. Both endpoints are excluded from the API versioning scheme and may not be removed or renamed.

12.10 Cross-Origin Resource Sharing

The API sets the following CORS headers on all responses:

- `Access-Control-Allow-Origin: *`
- `Access-Control-Allow-Methods: GET, OPTIONS`
- `Access-Control-Max-Age: 86400`

This permits browser-based clients to access the API directly without a server-side proxy.

13 Verification, Test Coverage, and Formal Assurance

This section specifies the test suite that a conforming implementation of the `ISI v 1.0` pipeline must pass. The suite is organised into five categories: unit tests, property-based tests, golden-file tests, integration tests, and integrity-chain tests. Each test is designated **T-*n*** and mapped to the determinism invariant(s) (Section 10.2) it verifies.

13.1 Test Infrastructure

The test suite is executed via a standard test runner (e.g., `pytest`). All tests are deterministic: no test uses random seeds, network access, or system clock queries. Tests that exercise stochastic properties (e.g., permutation invariance) use a fixed, reproducible set of permutations drawn from a seeded pseudo-random generator whose seed is committed to the test codebase.

Test data consists of two reference datasets:

fixtures/reference_small.json

A synthetic dataset of five countries and six axes, with known closed-form solutions. Used for unit tests and property-based tests where exact expected values can be computed by hand.

fixtures/reference_eu27.json

The full EU-27 input dataset for vintage 2024. Used for golden-file tests and integration tests. The expected outputs are stored as golden files alongside the fixture.

13.2 Unit Tests

Unit tests verify individual functions in isolation. Each test provides a known input and asserts exact equality with the expected output.

Table 12. Unit test registry.

ID	Description	Invariant
T-01	HHI of a uniform share vector $(\frac{1}{n}, \dots, \frac{1}{n})$ equals $\frac{1}{n}$.	—
T-02	HHI of a degenerate share vector $(1, 0, \dots, 0)$ equals 1.	—
T-03	Dual-channel arithmetic mean: $0.5C^{(A)} + 0.5C^{(B)}$ matches expected value for three test cases.	—
T-04	Category-weighted Channel B (Equation (3)) matches hand-computed value for two sub-categories with known volumes.	—
T-05	Fuel average (Equation (4)) matches expected value when one, two, and three fuel categories are present.	—
T-06	Composite (Equation (5)) matches arithmetic mean of six known axis scores.	—
T-07	Missing-axis composite (Equation (6)) with $ \mathcal{A}_i = 4$ matches expected value.	D-10

Continued on next page

Table 12. Unit test registry (continued).

ID	Description	Invariant
T-08	Rounding: input "0.123456785" at precision 8 returns "0.12345678" (ties to even).	D-5
T-09	Rounding: input "0.123456795" at precision 8 returns "0.12345680" (ties to even).	D-5
T-10	Classification of boundary values: 0.50 maps to highly_concentrated; 0.49999999 maps to moderately_concentrated.	D-7
T-11	Classification of all four tier boundaries (0.15, 0.25, 0.50, and values just below each).	D-7
T-12	Scenario clamping: input axis score 0.95 with $\alpha = +0.20$ returns 1.0, not 1.14.	D-9
T-13	Scenario clamping: input axis score 0.03 with $\alpha = -0.20$ returns 0.024, not a negative value.	D-9
T-14	Ranking tie-break: two countries with identical composite scores are ordered by ISO alpha-2 code (ascending).	D-8
T-15	Canonical hash string for a known country record matches a pre-computed SHA-256 digest.	D-6

13.3 Property-Based Tests

Property-based tests verify algebraic properties that must hold for *any* valid input, not merely for specific test vectors. Inputs are generated from constrained domains using a combinatorial or exhaustive strategy over small state spaces.

Table 13. Property-based test registry.

ID	Property	Invariant
T-20	Unit-interval bound. For any valid share vector, $0 < C_i^{(\text{ch})} \leq 1$.	—
T-21	Share-vector normalisation. Input share vectors that do not sum to 1 (within $\epsilon = 10^{-9}$) are rejected by the validation layer.	—
T-22	Monotonicity of concentration. Shifting share mass from a small partner to the largest partner weakly increases the HHI.	—
T-23	Composite bounds. $0 < \text{ISI}_i \leq 1$ for all countries with at least one computable axis.	—
T-24	Classification monotonicity. If $s_1 > s_2$, then $\text{tier}(s_1) \geq \text{tier}(s_2)$ under the ordering $\text{highly} > \text{moderately} > \text{mildly} > \text{unconcentrated}$.	D-7
T-25	Ranking antisymmetry. If country i ranks above country j , then country j does not rank above country i .	D-8
T-26	Ranking totality. Every country in the snapshot receives exactly one rank. No rank is assigned to two countries unless their composite scores are identical (in which case the tie-break applies).	D-8

Continued on next page

Table 13. Property-based test registry (continued).

ID	Property	Invariant
T-27	Scenario bounds. For any valid baseline score and any $\alpha \in [-0.20, +0.20]$, the simulated score lies in $[0, 1]$.	D-9
T-28	Scenario identity. When $\alpha_a = 0$ for all a , the simulated scores equal the baseline scores exactly.	D-9
T-29	Rounding idempotency. $R_8(R_8(x)) = R_8(x)$ for all x in the domain.	D-5
T-30	Hash stability. The SHA-256 digest of a canonical string is invariant across invocations.	D-6

13.4 Golden-File Tests

Golden-file tests compare the full pipeline output against a pre-computed reference. They are the primary mechanism for verifying end-to-end determinism.

Table 14. Golden-file test registry.

ID	Description	Invariant
T-40	Small-dataset golden file. Pipeline output on <code>reference_small.json</code> matches the golden file byte-for-byte.	D-1, D-2
T-41	EU-27 golden file. Pipeline output on <code>reference_eu27.json</code> matches the golden file byte-for-byte. This test verifies the production dataset.	D-1, D-2

Continued on next page

Table 14. Golden-file test registry (continued).

ID	Description	Invariant
T-42	Idempotency. Running the pipeline twice on <code>reference_eu27.json</code> produces byte-identical output.	D-1
T-43	Row-permutation invariance. Running the pipeline on 10 distinct random permutations of the input rows of <code>reference_small.json</code> produces identical output each time.	D-4
T-44	Scenario golden file. Running the scenario engine with a fixed adjustment vector on a fixed country produces output matching a pre-computed golden file.	D-9

13.5 Integration Tests

Integration tests verify the end-to-end pipeline including data ingestion, validation, computation, serialisation, and integrity-chain construction.

Table 15. Integration test registry.

ID	Description	Invariant
T-50	Snapshot completeness. The materialised snapshot directory contains all mandatory files listed in Section 11.2.	—
T-51	Manifest consistency. Every file listed in <code>MANIFEST.json</code> exists in the directory, and its SHA-256 hash matches the recorded value.	D-6

Continued on next page

Table 15. Integration test registry (continued).

ID	Description	Invariant
T-52	Hash-summary consistency. Recomputing all country hashes from <code>country_scores.csv</code> yields the same values as <code>HASH_SUMMARY.json</code> .	D-6
T-53	Snapshot-hash consistency. The snapshot hash H_t recomputed from sorted country hashes matches the value in <code>HASH_SUMMARY.json</code> .	D-6
T-54	Signature verification. The Ed25519 signature in <code>SIGNATURE.json</code> verifies against H_t using the published public key.	—
T-55	CSV schema compliance. All CSV output files conform to the column schemas specified in Section 11.3.	—
T-56	JSON schema compliance. The methodology registry and snapshot metadata files contain all mandatory keys listed in Sections 11.4 and 11.5.	—
T-57	Score-string format. All score strings in CSV and JSON output files match the regex <code>^(null -?[0-9]+\.[0-9]{8})\$</code> .	D-5
T-58	Warning propagation. Warning codes generated during computation appear in <code>warnings.csv</code> and are correctly associated with the respective country–axis pair.	—

Continued on next page

Table 15. Integration test registry (continued).

ID	Description	Invariant
T-59	Fallback-basis recording. The <code>fallback_basis.csv</code> file records one entry per country–axis pair, and the recorded basis matches the channel availability in the input data.	—

13.6 Integrity-Chain Tests

These tests specifically verify the cryptographic integrity chain in isolation.

Table 16. Integrity-chain test registry.

ID	Description	Invariant
T-60	Country-hash determinism. The same country record always produces the same SHA-256 digest.	D-6
T-61	Country-hash sensitivity. Changing a single decimal digit in one axis score produces a different country hash.	D-6
T-62	Snapshot-hash order dependence. The snapshot hash changes if the country-hash concatenation order is altered (verifying that lexicographic sorting is enforced).	D-6
T-63	Signature validity. A freshly computed signature over H_t verifies with the corresponding public key.	—

Continued on next page

Table 16. Integrity-chain test registry (continued).

ID	Description	Invariant
T-64	Tamper detection. Modifying one byte in any score file causes the manifest hash to differ from the recorded value, triggering a verification failure.	D-6
T-65	Signature rejection. An Ed25519 signature produced with a different private key fails verification against the published public key.	—

13.7 Coverage Requirements

A conforming implementation must satisfy the following minimum coverage thresholds:

Table 17. Minimum test-coverage requirements.

Module	Line coverage	Branch coverage
Concentration scoring	100%	100%
Cross-channel aggregation	100%	100%
Composite averaging	100%	100%
Classification	100%	100%
Rounding	100%	100%
Ranking	100%	100%
Scenario engine	100%	100%
Integrity chain (hashing)	100%	100%
Integrity chain (signing)	100%	100%
Serialisation (JSON/CSV)	≥ 95%	≥ 90%
Data ingestion / parsing	≥ 90%	≥ 85%
Validation	≥ 95%	≥ 90%

All core mathematical and classification modules require 100% line and branch coverage. This is non-negotiable: any untested branch in a scoring or classification function represents a potential determinism violation.

13.8 Regression Protocol

When a bug is discovered in the pipeline, the following protocol applies:

1. **Bug isolation.** A minimal reproducing test case is constructed and added to the test suite as a new test (**T-*n***).
2. **Fix implementation.** The bug is fixed in the pipeline codebase.
3. **Golden-file update.** If the fix changes the output for the reference datasets, the golden files are regenerated and the new expected output is committed alongside the fix.
4. **Version increment.** If the fix changes published output (axis scores, composite scores, classifications, or hashes), a minor or major version increment is triggered under the governance protocol (Section 9).
5. **Non-regression gate.** The full test suite must pass before the fix is merged. No test may be deleted to accommodate a fix.

13.9 Continuous Integration

The test suite runs automatically on every commit to the main branch and on every pull request. The CI pipeline enforces:

1. All tests pass (zero failures, zero errors).
2. Coverage thresholds (Table 17) are met.
3. No new static-analysis warnings are introduced (linting via `ruff` or equivalent).
4. No new type-checking errors are introduced (type checking via `mypy` in strict mode or equivalent).

A pull request that fails any of these gates cannot be merged.

13.10 Formal Assurance Statement

The test suite does not constitute a formal proof of correctness. It provides a high-confidence empirical assurance that the pipeline conforms to its specification. The assurance is bounded by the coverage and diversity of the test inputs. Specifically:

- Property-based tests (**T-20** through **T-30**) verify algebraic invariants over parameterised input spaces, providing stronger guarantees than fixed test vectors alone.
- Golden-file tests (**T-40** through **T-44**) provide byte-level regression protection for the complete pipeline.
- Integrity-chain tests (**T-60** through **T-65**) verify the cryptographic binding between scores and published hashes.

Formal verification (e.g., proof-carrying code, model checking) is not applied in v 1.0 but is identified as a candidate for future methodology versions where the state space and cost of verification warrant it.

14 Threat Model and Trust Assumptions

This section defines the adversary model, security goals, trust anchors, and key-management policy governing the ISI integrity system. The model is scoped to the computational pipeline, snapshot artefacts, and serving layer. Raw-data veracity is explicitly out of scope.

Normative keywords (SHALL, SHALL NOT, MUST, MUST NOT, MAY, RECOMMENDED) are used in the sense defined by Bradner [5] throughout this section.

14.1 Adversary Classes

The following adversary classes are considered. Each class is designated **A-*n*** and is referenced in the mitigation analysis below.

- A-1 Post-materialisation file tampering.** An adversary modifies one or more files in a published snapshot directory after the snapshot has been signed. The attack surface includes score files, the methodology registry, and metadata files. The adversary does not possess the signing key.
- A-2 Partial snapshot substitution.** An adversary replaces a subset of files in a snapshot directory with files from a different vintage or methodology version while leaving the remaining files intact. The goal is to produce a snapshot that passes superficial inspection but contains inconsistent data.
- A-3 API response manipulation.** A network-level adversary intercepts and modifies API responses between the serving infrastructure and the client. This includes modification of JSON payloads, HTTP headers, and ETag values.
- A-4 Registry mutation.** An adversary with write access to the deployment environment modifies the `methodology_registry.json` to alter parameter values (e.g., classification thresholds, rounding precision) without recomputing the affected scores.
- A-5 Floating-point drift.** A non-conforming implementation produces scores that differ from the canonical output due to platform-specific floating-point behaviour (extended precision, FMA contraction, non-IEEE rounding mode). The divergence is unintentional but produces hash mismatches and potentially different classifications.
- A-6 Key compromise.** The Ed25519 signing key is obtained by an adversary, enabling production of valid signatures over fabricated snapshots.

14.2 Security Goals

The integrity system SHALL provide the following guarantees:

- G-1 Snapshot integrity.** Any modification to any file within a signed snapshot SHALL be detectable by a verifier with access to the public key and the snapshot's `SIGNATURE.json`. This goal mitigates **A-1** and **A-2**.

- G-2 Deterministic reproducibility.** Any conforming implementation that executes the ISI pipeline on identical inputs and identical methodology parameters SHALL produce byte-identical output, including identical hash digests and an identical signature preimage. This goal mitigates **A-5**.
- G-3 Tamper detectability.** Verification of a snapshot SHALL require only the snapshot directory, the public key, and the verification algorithm. No network access, database query, or access to the computation environment SHALL be required. A failed verification SHALL produce an unambiguous rejection.
- G-4 Non-repudiation.** A validly signed snapshot constitutes a binding commitment by the custodian (Section 9.1) to the scores, classifications, and rankings contained therein. The custodian SHALL NOT deny authorship of a snapshot bearing a valid signature under a key they have published.

14.3 Out-of-Scope Threats

The following threats are explicitly excluded from the security model:

1. **Raw data falsification.** The integrity chain authenticates the *computation*, not the veracity of the underlying source data. If a statistical agency publishes erroneous trade data, the ISI pipeline will faithfully compute and sign a snapshot reflecting that error. Source-data validation is the responsibility of the data providers, not of the ISI integrity system.
2. **Upstream statistical bias.** Systematic biases in source-data collection methodologies (e.g., under-reporting of certain trade flows) are not detectable or correctable by the integrity chain.
3. **Compromised data providers.** If a data provider is coerced or incentivised to publish misleading data, the ISI system has no mechanism to detect or correct the resulting distortion. The integrity chain guarantees that the published snapshot correctly reflects the computation; it does not guarantee that the inputs to the computation are truthful.
4. **Side-channel attacks on the computation environment.** Timing attacks, power analysis, and electromagnetic emanation attacks against the hardware executing the pipeline are not considered. The pipeline is not a cryptographic primitive and does not process secret inputs.
5. **Denial of service.** Availability of the API serving layer is an operational concern governed by infrastructure policy, not by this specification. Rate limiting (Section 12.5) provides a baseline defence but does not constitute a formal availability guarantee.

14.4 Trust Anchors

The security model rests on the following trust assumptions. Violation of any assumption invalidates the corresponding security goal.

- TA-1 Public key distribution.** The Ed25519 public key is distributed through an authenticated channel (e.g., the project repository, a TLS-protected webpage, or a published document

bearing an institutional identifier). Verifiers **MUST** obtain the public key from a trusted source. If the public key is substituted, **G-1** and **G-4** are void.

- TA-2 Append-only registry.** The snapshot’s methodology-registry file is append-only at the deployment level. Existing entries **SHALL NOT** be modified or deleted. New entries **MAY** be appended for new vintages or methodology versions. This assumption underpins **G-1**.
- TA-3 Offline verification path.** A verifier with access to the snapshot directory and the public key **SHALL** be able to perform full integrity verification without network connectivity. The verification algorithm is specified in Section 7.6 and is reproducible from this document alone.
- TA-4 Signing-key confidentiality.** The Ed25519 private key is held exclusively by the custodian (Section 9.1) and is not shared with data providers, API operators, or third parties. This assumption underpins **G-4**.

14.5 Mitigation Matrix

Table 18 maps each adversary class to the integrity mechanism that mitigates it and the security goal that is preserved.

Table 18. Adversary-class mitigation matrix.

Adversary	Mechanism	Detection method	Goal
A-1	SHA-256 + Ed25519	Hash mismatch on tampered file; signature verification failure.	G-1
A-2	Manifest binding (C-3)	Manifest hash covers all files; substituted file produces hash mismatch.	G-1
A-3	ETag + offline verification	Client recomputes hashes from downloaded snapshot; TLS protects transport.	G-3
A-4	Registry in manifest	Registry file is hashed in manifest; mutation invalidates snapshot hash.	G-1
A-5	Determinism invariants (D-1–D-10)	Golden-file and cross-platform tests detect drift.	G-2
A-6	Key rotation protocol	Revocation + re-signing with new key; see Section 14.6.	G-4

14.6 Key Rotation and Revocation Policy

The following normative rules govern the Ed25519 signing key lifecycle:

1. The custodian **SHALL** generate a new Ed25519 key pair when the methodology undergoes a major-version increment.
2. The custodian **MAY** generate a new key pair at any time for operational reasons (e.g., personnel change, infrastructure migration).

3. Upon key rotation, the custodian SHALL publish the new public key through the same authenticated channels used for the previous key (TA-1).
4. The previous public key SHALL remain published and labelled as “superseded” with an effective date. Removal of a superseded public key is prohibited: verifiers of historical snapshots require access to the key that was active at the time of signing.
5. Snapshots signed with a superseded key SHALL NOT be re-signed with the new key. Each snapshot’s signature is bound to the key that was active at the time of publication. Re-signing would violate the immutability contract (Section 11.6).
6. In the event of suspected or confirmed key compromise (A-6), the custodian SHALL:
 - (a) Immediately revoke the compromised key by publishing a signed revocation notice using the compromised key (if still in possession) or via an authenticated institutional channel.
 - (b) Generate and publish a new key pair.
 - (c) Audit all snapshots signed with the compromised key and publish an integrity report identifying any snapshots whose authenticity cannot be confirmed.
 - (d) Re-sign affected snapshots with the new key only if the custodian can independently verify that the snapshot contents are authentic. Re-signed snapshots are published in an addendum directory alongside the original (Section 11.6).
7. Key pairs SHALL be generated using a cryptographically secure random number generator conforming to the requirements of Josefsson and Liusvaara [6]. The private key SHALL be stored in an access-controlled environment with audit logging.

14.7 Security Guarantees and Cryptographic Assumptions

The integrity system rests on two standard cryptographic hardness assumptions. If both hold, the security goals stated in Section 14.2 are satisfied.

Assumption CR (Collision resistance). SHA-256 is collision-resistant: no computationally bounded adversary can find distinct inputs $x \neq x'$ such that $\text{SHA-256}(x) = \text{SHA-256}(x')$. Under this assumption, any modification to any axis score, composite score, classification label, country ordering, warning code, or serialised field in a country record SHALL produce a different country hash h_i and therefore a different snapshot hash H_t .

Assumption EU (Existential unforgeability). Ed25519 is existentially unforgeable under chosen-message attack: no computationally bounded adversary, given the public key and access to a signing oracle, can produce a valid signature over a message not previously signed. Under this assumption, a valid signature σ_t over H_t constitutes proof that the custodian authorised the snapshot.

Formal integrity guarantee. Let \mathcal{S} be a published snapshot with signature

$$\sigma_t = \text{Ed25519-Sign}(sk, H_t).$$

Under Assumptions CR and EU, the following holds:

If a verifier possesses the authentic public key and executes the verification procedure of Section 7.6, then any post-materialisation modification to any file in \mathcal{S} SHALL cause verification to fail.

This guarantee covers all adversary classes **A-1** through **A-4**. It does not cover **A-5** (floating-point drift), which is mitigated by the determinism invariants (**D-1** through **D-10**), or **A-6** (key compromise), which is mitigated by the key-rotation protocol (Section 14.6).

Trust assumptions restated. The guarantee above is conditional on:

1. Public key authenticity (**TA-1**).
2. Raw data provider honesty is *not* assumed; the guarantee covers computation integrity, not input veracity.
3. The offline verification path (**TA-3**) is sufficient for independent validation; no trust in the serving infrastructure is required.

15 Formal Computational State Model

This section defines the ISI computation pipeline as a deterministic finite state machine. The model enumerates every intermediate state, every transition function, and every irreversibility constraint. Conforming implementations SHALL realise the state transitions in the order specified below; no reordering, merging, or elision of states is permitted.

15.1 State Definitions

The pipeline comprises ten discrete states, designated **S0** through **S9**. Each state is a typed artefact; the transition from state n to state $n + 1$ is a pure function consuming only the artefact at state n (and, where noted, the methodology-parameter vector θ).

- S0** **Raw data.** The unprocessed source files as received from statistical agencies. Format: heterogeneous (CSV, SDMX, JSON). No computational transformation has been applied. State S0 is the sole external input to the pipeline.
- S1** **Validated data.** Source files ingested, parsed, and validated against the quality controls of Section 2.5. Records failing validation are rejected. Output: a normalised tabular dataset with uniform column schema, country codes in ISO 3166-1 alpha-2, and numeric fields in native floating-point representation.
- S2** **Channel concentration vectors.** Per-country, per-axis, per-channel partner-share vectors and the corresponding HHI scores computed via Equation (1). Output: a set of $(i, a, c, \text{HHI}_{i,a,c})$ tuples for every country i , axis a , and channel c .
- S3** **Axis-level scores.** Channel scores aggregated to axis level via the aggregation variants specified in Section 4: dual-channel averaging (Equation (2)), category-weighted averaging (Equation (3)), or single-channel fuel averaging (Equation (4)). Output: a set of $(i, a, M_i^{(a)})$ tuples. Scores at this stage are in full binary64 precision.
- S4** **Composite score.** The arithmetic mean of computable axis scores (Equation (5)). Output: a set of (i, ISI_i) pairs in full binary64 precision.
- S5** **Rounded and classified country object.** All scores (axis and composite) rounded to eight decimal places (round-half-to-even) using an arbitrary-precision decimal library. The classification function (Section 3.8) is applied to the rounded composite. The ranking function (Section 5.3) produces ordinal ranks. Output: a fully populated country record containing rounded scores, classification label, rank, warning codes, and computable-axis count.
- S6** **Country hash.** The canonical serialisation of each country record (Section 7.2) is hashed with SHA-256 to produce h_i . Output: a set of (i, h_i) pairs.
- S7** **Snapshot hash.** Country hashes concatenated in ISO alpha-2 lexicographic order, combined with the manifest hash, and hashed with SHA-256 to produce H_t (Section 7.3). Output: a single 256-bit digest.

- S8 Signed snapshot.** The snapshot hash is signed with Ed25519 (Section 7.4) to produce σ_t . The snapshot directory is materialised on disk (Section 11). Output: the complete snapshot directory including all score files, integrity files, and the signature.
- S9 Served API artefact.** The signed snapshot is loaded into the serving layer (Section 12). API responses are read-only projections of the snapshot. No transformation, re-computation, or re-serialisation occurs at serving time.

15.2 Transition Functions

Each transition is designated **T n** and maps from state **S(n-1)** to state **S n** . All transitions are pure functions: they read only their input state and the methodology-parameter vector θ ; they produce no side effects.

- T1** $f_1 : (S0, \theta) \rightarrow S1$.
Ingestion and validation. Applies schema enforcement, type coercion, country-code normalisation, and quality-control rejection rules (Sections 2.4 and 2.5).
- T2** $f_2 : (S1, \theta) \rightarrow S2$.
Share-vector construction and HHI computation (Equation (1)). Partner shares are normalised to sum to 1.0 per country–axis–channel group. Summation follows the canonical accumulation order specified in Section 10.3.
- T3** $f_3 : (S2, \theta) \rightarrow S3$.
Cross-channel aggregation. Applies the axis-specific aggregation variant (Section 4). Output scores are in full binary64 precision.
- T4** $f_4 : (S3, \theta) \rightarrow S4$.
Composite averaging (Equation (5)). The divisor is the number of computable axes $|\mathcal{A}_i|$, not a fixed constant.
- T5** $f_5 : (S4, \theta) \rightarrow S5$.
Rounding, classification, and ranking. This is the **rounding gate**: the irreversible precision-reduction step. After this transition, all scores are fixed-point 8-decimal values and SHALL NOT revert to binary64 representation.
- T6** $f_6 : S5 \rightarrow S6$.
Country hashing. Canonical serialisation of the country record followed by SHA-256 (Section 7.2). This is the **hashing gate**: once a country hash is computed, the underlying score values are cryptographically committed.
- T7** $f_7 : S6 \rightarrow S7$.
Snapshot hash construction. Concatenation of ordered country hashes and manifest hash, followed by SHA-256 (Section 7.3).
- T8** $f_8 : S7 \rightarrow S8$.
Signing and materialisation. Ed25519 signature over H_t , followed by file-system materialisation of the complete snapshot directory.

T9 $f_9 : S_8 \rightarrow S_9$.
Loading into the serving layer. Read-only memory mapping of the snapshot directory.
No mutation of the snapshot is permitted during or after this transition.

15.3 Irreversibility Points

Two transitions impose irreversible commitments on the pipeline state. After each irreversibility point, the preceding full-precision representation is discarded and SHALL NOT be recoverable from the published artefacts.

T5 (Rounding gate)

After transition **T5**, all published scores exist only in rounded 8-decimal form. The full binary64 intermediates are not serialised, not hashed, and not recoverable. Any downstream computation (classification, hashing, serving) operates exclusively on rounded values. Invariant **D-5** guarantees that no further rounding is required.

T6 (Hashing gate)

After transition **T6**, the content of each country record is cryptographically committed via SHA-256. Any modification to a rounded score, classification label, or warning code after **T6** would produce a different country hash and therefore a different snapshot hash, invalidating the signature.

The two gates partition the pipeline into three zones:

1. **Pre-rounding zone** (S0–S4). Full binary64 precision. Scores may be recomputed freely.
2. **Post-rounding / pre-hashing zone** (S5). Rounded scores exist but are not yet cryptographically committed. Recomputation is permitted only if the rounding step is re-executed.
3. **Post-hashing zone** (S6–S9). Scores are committed. Modification is prohibited (Section 11.6).

15.4 State Mutation Prohibition

The pipeline is a strict feed-forward computation. The following mutation rules SHALL be enforced:

1. No transition function f_n MAY modify the artefact at state $S_{(n-1)}$. Transitions are pure functions that *read* input states and *produce* output states.
2. After transition **T8** (signing), the snapshot directory SHALL NOT be modified by any subsequent operation, including **T9** (serving). The serving layer operates on a read-only view of the materialised snapshot.
3. No transition function MAY write to a global mutable variable, file, or database that is read by a later transition. All inter-state communication is through the explicitly typed artefacts defined in Section 15.1.

15.5 Scenario Engine as Non-State-Altering Branch

The scenario engine (Section 6) operates as a **read-only branch** from state **S5**. It consumes the rounded baseline scores and the user-supplied adjustment vector $(\alpha_1, \dots, \alpha_6)$, applies the multiplicative model (Equation (7)), recomputes the composite and rank, and returns the result to the caller.

The scenario engine SHALL NOT:

- Modify any file in the snapshot directory.
- Write to the methodology registry.
- Produce or modify country hashes, snapshot hashes, or signatures.
- Persist scenario results in any durable store that is visible to the canonical pipeline.

Scenario outputs are **ephemeral**: they exist only in the API response and are not covered by the integrity chain. Scenario responses SHALL NOT include a signature or hash digest. Any downstream consumer of scenario data MUST treat it as unsigned and non-authoritative.

Formally, the scenario engine defines a side-branch:

$$f_{\text{scenario}} : (S5, \vec{\alpha}) \rightarrow S5'_{\text{ephemeral}}$$

where $S5'_{\text{ephemeral}}$ is a transient artefact that is never committed to **S6** or any subsequent state. The canonical state path $S0 \rightarrow \dots \rightarrow S9$ is unaffected.

15.6 Deterministic Transition Constraints

Every transition function f_n SHALL satisfy the following properties:

1. **Purity.** f_n depends only on its input state and θ . No hidden input (clock, locale, environment variable, random seed) is consumed. This is a restatement of invariants **D-1** through **D-3**.
2. **Totality.** f_n is defined for all valid inputs. Partial failures (e.g., a missing axis score) are handled by the missing-axis protocol (Section 5.2), not by undefined behaviour.
3. **Convergence.** f_n terminates in finite time for all valid inputs. No unbounded iteration or recursion is permitted.
4. **Idempotence of read operations.** Reading from a state artefact does not modify it. This is trivially satisfied by file-system reads but must be explicitly enforced for in-memory data structures that might use lazy evaluation or memoisation with side effects.
5. **Canonical ordering.** Where the transition iterates over a collection (partner shares, country records, axis scores), the iteration order SHALL be the canonical order defined in Section 10.3 and Section 10.4.

15.7 Axis-Completeness Invariant

The pipeline enforces a formal completeness requirement at transition **T5**. Let $A_{\text{required}} = 6$ denote the number of axes defined in the methodology, and let \mathcal{A}_i denote the set of axes with valid rounded scores for country i at state **S5**.

Completeness predicate.

$$\text{complete}(i) \iff |\mathcal{A}_i| = A_{\text{required}}$$

Materialisation rule. The following rules SHALL be enforced at transition **T5**:

1. If $|\mathcal{A}_i| = A_{\text{required}}$, the country record is complete and proceeds to **T6** (country hashing).
2. If $4 \leq |\mathcal{A}_i| < A_{\text{required}}$, the composite is computed over the available axes with divisor $|\mathcal{A}_i|$ (Equation (6)). The country record is flagged with warning code `W-COV` and proceeds to **T6**. The hash preimage includes the reduced axis set.
3. If $|\mathcal{A}_i| < 4$, the composite SHALL NOT be materialised. The country SHALL be excluded from the ranked output. No country hash SHALL be computed. The country is recorded in `warnings.csv` with code `W-EXCL`.

This rule ensures that no country enters the integrity chain (**S6–S8**) without a well-defined composite score and a complete or explicitly flagged axis set.

15.8 State Diagram Summary

The following notation summarises the state machine. Each arrow is labelled with its transition function and the section that specifies it.

```
S0 --[T1: ingest/validate (sec 2.4)]--> S1
S1 --[T2: HHI computation (sec 3.2)]--> S2
S2 --[T3: axis aggregation (sec 3.3)]--> S3
S3 --[T4: composite mean (sec 3.6)]--> S4
S4 --[T5: round+classify (sec 3.7)]--> S5 *ROUNDING GATE*
S5 --[T6: country hash (sec 7.2)]--> S6 *HASHING GATE*
S6 --[T7: snapshot hash (sec 7.3)]--> S7
S7 --[T8: sign+materialise (sec 7.4)]--> S8
S8 --[T9: serve (sec 12)]--> S9

S5 --[f_scenario (sec 6)]--> S5'_ephemeral (branch)
```

No backward edge exists. The graph is a directed acyclic path with a single read-only branch at **S5**. This structure guarantees that every state is reachable from **S0** by a unique path and that no cyclic dependency can arise.

16 System Architecture Overview

The ISI system is organised into four layers. Each layer encapsulates a distinct set of responsibilities and invariants. The layers are strictly ordered: each layer depends only on the layer below it; no upward or circular dependency exists.

Layer 1 Mathematical Layer

Scope. Concentration measurement, aggregation, and composite construction.

Core operations. Herfindahl–Hirschman concentration (Equation (1)), cross-channel aggregation (Equations (2) to (4)), arithmetic composite mean (Equation (5)), four-tier classification (Table 2).

Invariants. All formulas are fixed for the lifetime of a major version. Axis weights are equal. Rounding precision is eight decimal places, round-half-to-even. No stochastic or discretionary element enters the computation.

State coverage. S0–S4 in the formal state model (Section 15).

Layer 2 Deterministic Computation Layer

Scope. Platform-independent determinism, canonical serialisation, and prohibited-operation enforcement.

Core operations. Rounding gate (T5), canonical UTF-8/LF/lexicographic-JSON serialisation (Section 10.4), country-code ordering, locale independence, prohibited-operation enforcement (Section 10.5).

Invariants. D-1 through D-10 (Section 10.2). IEEE 754 binary64 arithmetic (Section 10.3). Hash-breakage clause: any serialisation deviation changes the snapshot hash.

State coverage. Transition T5 (rounding gate) at boundary S4 → S5.

Layer 3 Integrity Layer

Scope. Cryptographic binding, tamper detection, and non-repudiation.

Core operations. Country-level SHA-256 hashing (T6), snapshot hash construction (T7), Ed25519 signing (T8), manifest binding (C-3), hash cascade (Section 11.9).

Invariants. Collision resistance and preimage resistance of SHA-256. Existential unforgeability of Ed25519. Immutable Snapshot Law (Section 11.6). Tamper propagation (C-5).

State coverage. S5–S8 (post-rounding through signed snapshot).

Security model. Adversary classes A-1 through A-6, security goals G-1 through G-4 (Section 14).

Layer 4 Governance Layer

Scope. Versioning law, registry management, immutability policy, key lifecycle, and publication protocol.

Core operations. Three-part version numbering (Section 9.2), append-only registry (TA-2), change protocol (Section 9.3), key rotation (Section 14.6), archival policy (Section 9.6).

Invariants. Version-definition immutability. No retroactive recomputation. Published snapshots are permanent. Superseded keys remain accessible.

State coverage. Post-S8 policy constraints and S9 (serving).

The four layers collectively ensure that the ISI system is mathematically specified (Layer 1), deterministically reproducible (Layer 2), cryptographically verifiable (Layer 3), and institutionally governed (Layer 4). No layer may be removed or weakened without a major-version increment.

References

- [1] Savina Gygli, Florian Haelg, Niklas Potrafke, and Jan-Egbert Sturm. The KOF globalisation index—revisited. KOF Working Paper 439, KOF Swiss Economic Institute, ETH Zurich, 2019.
- [2] UNCTAD. Productive capacities index: Methodological note. Technical note, United Nations Conference on Trade and Development, 2023. URL <https://unctad.org/topic/least-developed-countries/productive-capacities-index>.
- [3] US Department of Justice and Federal Trade Commission. Merger guidelines. Guidelines, US Department of Justice, 2023. URL <https://www.justice.gov/atr/merger-guidelines-2023>. Revised 18 December 2023.
- [4] Daniel J. Bernstein, Niels Duif, Tanja Lange, Peter Schwabe, and Bo-Yin Yang. High-speed high-security signatures. *Journal of Cryptographic Engineering*, 2(2):77–89, 2012. doi: 10.1007/s13389-012-0027-1.
- [5] Scott Bradner. Key words for use in RFCs to indicate requirement levels. RFC 2119, March 1997. URL <https://www.rfc-editor.org/rfc/rfc2119>.
- [6] S. Josefsson and I. Liusvaara. Edwards-curve digital signature algorithm (EdDSA). RFC 8032, Internet Engineering Task Force, 2017.

A Axis Registry

Table 19 provides the complete axis registry in tabular form.

Table 19. Complete axis registry.

Ax.	Domain	Source	Ch. A	Ch. B	Special
1	Financial sov.	BIS LBS / IMF CPIS	Inward claims	Portfolio debt	Fallback basis
2	Energy dep.	Eurostat nrg_ti_*	Per-fuel HHI (no channels)		Fuel average
3	Tech./semi.	Comext ds-045409	Aggregate HHI	Category-wtd	HS 8541+8542
4	Defence ind.	SIPRI ATD	Aggregate HHI	Capability-wtd	6-yr window
5	Critical inp.	Comext CN8	Aggregate HHI	Material-wtd	CRM Act scope
6	Logistics	Eurostat transport	Mode HHI	Partner HHI	Tonnage-wtd

B Classification Thresholds

Table 20. Classification thresholds (reproduced from Table 2).

Tier	Composite range	Label
1	$ISI_i \geq 0.50$	highly_concentrated
2	$0.25 \leq ISI_i < 0.50$	moderately_concentrated
3	$0.15 \leq ISI_i < 0.25$	mildly_concentrated
4	$ISI_i < 0.15$	unconcentrated

The thresholds represent a proportional transposition of the US DOJ/FTC Horizontal Merger Guidelines HHI bands [3]—which classify markets as highly concentrated (HHI above 2500), moderately concentrated (HHI 1500–2500), and unconcentrated (HHI below 1500) on the 0–10 000 scale—to the fractional scale, with an additional tier boundary at 0.15 introduced to separate mildly concentrated from unconcentrated states.

C Warning Code Registry

Table 21 enumerates all warning codes defined in ISI v 1.0.

Table 21. Warning code registry.

Code	Scope	Description
W-COV	Country	Fewer than four axes computable; country excluded from ranked output.
W-ZER	Channel	Zero recorded flow; channel or axis score set to null.

Continued on next page

Table 21. Warning code registry (continued).

Code	Scope	Description
W-LAG	Axis	Most recent data predates vintage year by more than two years.
W-DEF	Axis 4	Fewer than three transfer events in the six-year window.
W-FIN	Axis 1	Country absent from BIS reporting population; Channel A score null.
W-CPI	Axis 1	Country absent from IMF CPIS; Channel B score null.